Welcome to the Curriculum for Web Content Accessibility Guidelines 1.0

This file: last modified March 17, 2000

Start here ▶

If you have comments about the content or design of the example sets please direct them to the Education and Outreach mailing list.

Zipped archives available.

Note: the zipped Curriculum slide set contains the html and supporting images necessary to display most of the examples. However, the multimedia examples for audio captioning and video description are not included and are zipped separately.

- Download the Curriculum slide sets in ZIP format.
  Approximately 1MB: (without multimedia)
  Date of archive: March 17, 2000

For instructions on how to use the multimedia files, follow the steps in the Multimedia Software instructions file.

- Download the SMIL/Realmovie files for 'Car' in ZIP format.
  Approximately 2.3 MB (date: November 29, 1999)
- Download the SMIL/Realmovie files for 'Elevator' in ZIP format.

Approximately 5.4 MB (date: November 29, 1999)

- Download the QuickTime movie 'Catch" in ZIP format.
  Approximately 2.4 MB (date: November 29, 1999)

This curriculum was prepared for the WAI by
Chuck Letourneau,
Starling Access Services
and
Geoff Freed,
National Center for Accessible Media

Introduction

Curriculum for Web Content Accessibility Guidelines 1.0: Course Outline

Please read this page before continuing with the curriculum.

The Curriculum

This curriculum comprises:

1. An Introductory section we will refer to as "The Introduction Set",
2. Guidelines for Web Content Accessibility, otherwise known as "The Guideline Set",
3. Checkpoints for meeting the Guideline requirements, or "The Checkpoint Set",
4. and Examples for implementing the Checkpoints, or "The Example Set".

The Introduction set covers such topics as:

- Appropriate use -- some cautions and disclaimers -- IMPORTANT!
- Your basic curriculum toolkit -- suggested software and hardware to help you get the most out of the curriculum
- How to use the curriculm -- suggestions for classroom use or for self study
- Navigation -- a description of the general design and navigation features of the curriculum
- A brief description of the World Wide Web Consortium (the W3C) and the Web Accessibility Initiative (the WAI)
- copyright information

The Guideline Set

The Guideline set presents a version of the W3C Recommendation "Web Content Accessibiliy Guidelines" (See the note about compliance, below). This set presents each guideline in separately. It explains why following this guideline is important to the accessibility of a web site, a web page or a web application. Each guideline slide links to a subsequent set of slides that highlight the "checkpoints" to consider when applying a guideline.

The Checkpoint Set

Each guideline refers to one or more "checkpoints" you need to consider when attempting to meet a guideline. In the Checkpoint Set, each slide shows the numbered checkpoints and the priority assigned to it. Each checkpoint then links to one or more example slides.

The Example Set

For each checkpoint, there are one or more pages of examples. The slides in the Example Set show how to implement each checkpoint, and where possible demonstrate what the result will look like in common use. In some cases numerous examples are needed to illustrate the techniques that will satisfy a checkpoint.

Please note that in all questions of compliance, the Recommendation is considered the definitive reference.

Links to introductory topics
[Welcome page | Course Outline | Appropriate use | Your basic curriculum toolkit | Using the curriculum | Navigation | The W3C | The WAI | Copyright]

Introduction Guidelines Checkpoints Examples ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Introduction

Introduction
Slide 2 of 18

Web Content Accessibility Guidelines Curriculum

> "The power of the Web is in its universality.
> Access by everyone regardless of disability is an essential aspect."
>
> **Tim Berners-Lee**, W3C Director and Inventor of the World Wide Web

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Introduction

Appropriate Use Notice

Introduction
Slide 3 of 18

- This curriculum is a relatively stable presentation. Some examples are still being refined as the Web Content Accessibility Working Group continues its investigations into appropriate techniques.
- These slides are being made available as a service to page authors, students and educators by the Education and Outreach Working Group of the Web Accessibility Initiative.
- Any use of these materials is at the user's risk.
- Please observe the W3C copyright conditions
- Please read the cautionary notes on the following pages as well.

Next section: W3C Recommendations.

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Introduction

The W3C Recommendations

Introduction
Slide 4 of 18

While the curriculum may be used separately from other W3C products, it does not stand alone:

- The W3C Web Content Accessibility Guidelines 1.0 is the definitive source for the material presented here. Accompanying the Guidelines are the Techniques for Web Content Accessibility Guidelines 1.0
- The W3C HTML 4.0 Recommendation is the definitive source for questions about HTML.
- The W3C Cascading Style Sheets (CSS1) Level 1 Specification
- The W3C Cascading Style Sheets, level 2 (CSS2) Specification

Always check the W3C Technical Reports and Publications page for the latest releases.

Next section: Our use of markup.

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Introduction

◀ ▶

How our use of markup may effect your use of these slides.

This curriculum makes extensive use of HTML 4.0 and CSS 1.0 techniques, but as of February 2000:

- few browsers fully support HTML 4.0 (although more do with each new release),
- there are still significant differences between browsers in their support for and rendering of Cascading Style Sheets (CSS) 1.0,
- few browsers fully support CSS 2.0.

Since a major goal of this curriculum is to show authors the benefits of using the new techniques:

- some examples have been simulated using current techniques that attempt to show what a future browser will display, and
- some examples that use HTML 4.0 and CSS 1.0 are only known to display properly on a few browsers (e.g. - and this list is not exhaustive - Microsoft Internet Explorer 4.01 or later or Netscape Navigator 4.0, Opera 3.51 (with optional plug-ins)).

In some cases the browser you are using will not support the particular markup we have used to illustrate an example. In those instances, we have made an attempt to ensure that what you will see (or hear) will still make sense in context and explain what is supposed to be happening.

The most important thing to remember is that Web browsing tools and assistive technology are constantly evolving. We hope the "next" release of every tool will support more accessibility features and W3C recommendations than the current release.

Next section is: Your basic curriculum toolkit.

Links to introductory topics
[Welcome page | Course Outline | Appropriate use | Your basic curriculum toolkit | Using the curriculum | Navigation | The W3C | The WAI | Copyright]

Introduction Guidelines Checkpoints Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Introduction

Your basic curriculum toolkit.

Introduction
Slide 6 of 18

◀ ▶

In order to demonstrate the accessibility and appearance of the new features of HTML 4.0 and Cascading style sheets, we have used those techniques frequently throughout this curriculum. The authors have made every attempt to code examples in such a way that no important explanatory information is lost if you use older or less sophisticated technology. However, because of the design choices we made, appreciating the full effect of the examples will require you to:

- use an HTML 4.0 and CSS 1.0 aware browser to view the curriculum (e.g. the latest versions of Internet Explorer, Navigator, Opera)
- enable image loading, Java, and cascading style sheets in your browser
- use a display resolution of 800 x 600
- ensure your computer has an audio card and suitable speakers
- ensure you have downloaded and tested the various multimedia players required

Optional: an HTML editor (to code examples or test things on-the-fly), HTML 4 and CSS reference manuals (on hand to answer detailed questions about those topics).

Next section is: Using the curriculum...

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Introduction

Suggestions for use of this curriculum.

Introduction
Slide 7 of 18

◀ ▶

(If you have not already done so, please read the Course Outline for important information about the general design of the curriculum)

How you choose to use these slides will depend on your particular needs and situation. Here are some of the more likely methods:

- You are an educator training students in a classroom or workshop setting
  - online, using a computer connected to the Internet
  - offline, using a computer with the curriculum stored locally
  - offline, using material printed from the site (e.g on paper or overhead transparency)
- You are using the curriculum as a self-study tool (live)
- You want to use selected examples for reference while designing a page or site (live or printed for off-line use).

The following slides provide suggestions and tips that will make the most of this curriculum.

Next section is: Usage: in a classroom setting.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Introduction

Usage: in a classroom setting.

Introduction
Slide 8 of 18

◀ ▶

Before using this curriculum in a classroom setting, please review the disclaimer and copyright sections (slides 3 to 5 and slide 17, respectively). The authors strongly encourage you to begin any new training course with some background on the W3C and WAI (slides 13 to 16 in this Introduction set).

You can think of the Guideline, Checkpoint and Example sets as separate (but related) teaching modules.

1. Teaching only the Guideline set:
    - Appropriate for an overview presentation.
    - Explain the concepts of designing accessible pages.
    - Describe the rationale for each guideline.
    - No "drilling" down to other modules (just use the right arrow/next page navigation button to advance throught the module).
    - Lower level of technical knowledge required by audience.
    - Can be tailored to short presentations using the Table of Contents to select specific guidelines.

2. Teaching both the Guideline and Checkpoint sets:
    - Appropriate for a more detailed presentation.
    - The Guideline slide explains the concepts and rationale.
    - Then navigate to the Checkpoint slide and review checkpoints for that guideline.
    - Some knowledge of HTML is expected of the audience.
    - Can be tailored to shorter presentations by selecting topics of interest from Table of Contents.

3. Teaching the Guideline, Checkpoint and Example sets:
    - Appropriate for a very detailed presentation.
    - The Guideline slide explains the concepts and rationale.
    - Then navigate to the Checkpoint slide and review checkpoints for that guideline.

  - Then navigate to the Example slide(s) and review the example(s) for that checkpoint of the guideline.
  - Greater knowledge of HTML is expected of the audience.
  - Can be tailored to shorter presentations by selecting topics of interest from Table of Contents.

Next section is: Usage: Speaker's notes.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Introduction

Usage: Speaker's notes

Please note: Speaker's notes are NOT provided with the current slide sets. In most cases, all relevent comments have been merged with the content of the Example slides. A future revision of this curriculum may make use of speaker's notes. At that time, the following comments will apply:

- While you can link to the speaker's note for an example it might be distracting to remove the example slide from the class in order to read from the speaker's note
- We suggest you print the speaker's notes and refer to them as you discuss the corresponding example.

Next section is: Usage: Self-study...

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Introduction

Usage: self-study.

Introduction
Slide 10 of 18

◀ ▶

- Use the curriculum in any of the ways described for classroom use (Slide 8) , or,
- Use the Guideline, Checkpoint, or Example set's Tables of Content to look up specific topics for study.

Next section is: Usage: Reference...

Introduction Guidelines Checkpoints Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Introduction

Usage: reference.

The Web Content Accessibility Guideline Curriculum sections are designed to make it easy to find information about any guideline.

Use the Guideline Table of Contents to find the guideline that interests you. When you get to the guideline slide you want you can navigate sequentially through the guidelines or link to that guideline's checkpoints and examples.

Next section is: On-line navigation...

Links to introductory topics
[Welcome page | Course Outline | Appropriate use | Your basic curriculum toolkit | Using the curriculum | Navigation | The W3C | The WAI | Copyright]

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Introduction

On-line Navigation aids

Introduction
Slide 12 of 18

◀ ▶

The following general navigation aids will appear in the top right corner of almost every slide:

- a slide counter to tell you your relative position in the current set,
- the left-arrow button ◀ (or "Previous page." link) takes you to the previous slide in the current set,
- the right-arrow button ▶ (or "Next page." link) takes you to the next slide in the current set.
- NOTE: Some special supplementary slides will require you to use the BACK, GO BACK, or PREVIOUS PAGE function of your browser to return to the page that linked to them.

In the main body of a slide, links will be provided to take you to other pages of important or associated information (if required).

- from a guideline slide, the main body will contain a link to its associated checkpoint slide;
- a checkpoint slide includes links to examples and supporting information;
- checkpoint and example slides include a link near the end of the main body that will return you to the page 'Up one level' (e.g. If you are viewing the example for checkpoint 2.4, the 'Up one level' link will take you to the list of checkpoints for Guideline 2.);
- there is also a link to take you back to the top of the page so you can more easily find the previous and next page buttons.

In the current design, the bottom of each slide contains the following navigable items:

- A bar of buttons
  Introduction | Guidelines | Checkpoints | Examples
  will link you to the Table of Contents for the Introduction, Guideline, Checkpoint and Example sets respectively.

For browsers that support the ACCESSKEY attribute, you can press the appropriate "attention" key (it varies from browser to browser), then:

- "p" to go to the previous page
- "n" to go to the next page
- "i" to go to the Introduction Overview

- "g" to go to the Guidelines Overview
- "c" to go to the Checkpoints Overview
- "e" to go to the Examples Overview

Finally, each slide ends with the following standardized information and links:

- Links to information about the co-authors.
- The W3C logo with a link to the W3C home page.
- The WAI logo with a link to the WAI home page.
- A statement of copyright with a link to more detailed copyright information.

Next section is: About the W3C.

Links to introductory topics
[Welcome page | Course Outline | Appropriate use | Your basic curriculum toolkit | Using the curriculum | Navigation | The W3C |
The WAI | Copyright]

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Introduction

Introduction
Slide 13 of 18

◀ ▶

About the World Wide Web Consortium (W3C)

- created to lead the Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability
- an international industry consortium jointly hosted by the MIT Laboratory for Computer Science (LCS) in the USA, the National Institute for Research in Computer Science and Control (INRIA) in France and Keio University in Japan.
- more than 350 organizations are Members of the Consortium (February 2000)

This section continues...

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Introduction

The W3C, cont'd

Introduction
Slide 14 of 18

Services provided by the Consortium include:

- a repository of information about the World Wide Web for developers and users;
- reference code implementations to embody and promote standards;
- and various prototype and sample applications to demonstrate use of new technology.

Visit their website for more information about the World Wide Web Consortium.

See the list of current W3C Recommendations and other technical documents.

The W3C also provides validation services. Check the W3C home page for this and other useful services.

Next section is: About the WAI.

Links to introductory topics
[Welcome page | Course Outline | Appropriate use | Your basic curriculum toolkit | Using the curriculum | Navigation | The W3C | The WAI | Copyright]

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Introduction

Introduction
Slide 15 of 18

◀ ▶

About the Web Accessibility Initiative (WAI)

The WAI is pursuing accessibility of the Web through five primary areas of work:

- addressing accessibility issues in the technology of the Web;
- creating guidelines for browsers, authoring tools, and content creation;
- developing evaluation and validation tools for accessibility;
- conducting education and outreach;
- and tracking research and development.

This section continues...

Introduction Guidelines Checkpoints Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Introduction

The WAI International Program Office (IPO)

Introduction
Slide 16 of 18

◀ ▶

The IPO enables partnering and coordination among the many stake-holders in Web accessibility: industry, disability organizations, government, and research organizations, and is sponsored by:

- the US National Science Foundation
- the Department of Education's National Institute on Disability and Rehabilitation Research;
- the European Commission's TIDE Program,
- The Government of Canada,
- and W3C industry Members including IBM/Lotus Development Corporation, Microsoft Corporation and NCR.

Disability and research organizations on several continents also actively participate in the WAI.

Next section is: W3C Copyright information...

Links to introductory topics
[Welcome page | Course Outline | Appropriate use | Your basic curriculum toolkit | Using the curriculum | Navigation | The W3C | The WAI | Copyright]

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Introduction

W3C Copyright information

Introduction
Slide 17 of 18

Your interactions with this site are in accordance with our public and Member privacy statements.

Next section is: Let's begin...

Links to introductory topics
[Welcome page | Course Outline | Appropriate use | Your basic curriculum toolkit | Using the curriculum | Navigation | The W3C | The WAI | Copyright]

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Introduction

Let's begin...

You have finished the Introductory section of the WAI Web Content Accessibility Guideline Curriculum.

You can now start with the introduction slide for:

- The Guideline Set
- The Checkpoint Set
- The Example Set

or

- return to the Curriculum Welcome Page

Chuck Letourneau & Geoff Freed

Guideline

Introduction to The Guideline Set

Guideline
Slide 1 of 17

▶

The following pages are referred to as "The Guideline Set" in the introduction to the curriculum.

Each page in this set highlights one guideline. For consistency with the published source, the text of each guideline closely reflects that found in the Web Content Accessibility Guidelines V1.0.

Each guideline has a link to a page with one or more "checkpoints" that an author will consider when ensuring the accessibility of a page design.

Next slide: Guideline 1.

Introduction | Guidelines | Checkpoints | Examples ▶

Chuck Letourneau & Geoff Freed

Guideline
Slide 2 of 17

◀ ▶

Guideline

1. Provide equivalent alternatives to auditory and visual content.

Provide content that, when presented to the user, conveys essentially the same function or purpose as auditory or visual content.

Although some people cannot use images, movies, sounds, applets, etc. directly, they may still use pages that include equivalent information to the visual or auditory content. The equivalent information must serve the same purpose as the visual or auditory content. Thus, a text equivalent for an image of an upward arrow that links to a table of contents could be "Go to table of contents". In some cases, an equivalent should also describe the appearance of visual content (e.g., for complex charts, billboards, or diagrams) or the sound of auditory content (e.g., for audio samples used in education).

This guideline emphasizes the importance of providing text equivalents of non-text content (images, pre-recorded audio, video). The power of text equivalents lies in their capacity to be rendered in ways that are accessible to people from various disability groups using a variety of technologies. Text can be readily output to speech synthesizers and braille displays, and can be presented visually (in a variety of sizes) on computer displays and paper. Synthesized speech is critical for individuals who are blind and for many people with the reading difficulties that often accompany cognitive disabilities, learning disabilities, and deafness. Braille is essential for individuals who are both deaf and blind, as well as many individuals whose only sensory disability is blindness. Text displayed visually benefits users who are deaf as well as the majority of Web users.

Providing non-text equivalents (e.g., pictures, videos, and pre-recorded audio) of text is also beneficial to some users, especially nonreaders or people who have difficulty reading. In movies or visual presentations, visual action such as body language or other visual cues may not be accompanied by enough audio information to convey the same information. Unless verbal descriptions of this visual information are provided, people who cannot see (or look at) the visual content will not be able to perceive it.

Checkpoints for guideline 1.

Next slide: Guideline 2.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Guideline

2. Don't rely on color alone.

Guideline
Slide 3 of 17

◀ ▶

Ensure that text and graphics are understandable when viewed without color.

If color alone is used to convey information, people who cannot differentiate between certain colors and users with devices that have non-color or non-visual displays will not receive the information. When foreground and background colors are too close to the same hue, they may not provide sufficient contrast when viewed using monochrome displays or by people with different types of color deficits.

✓ Checkpoints for guideline 2.

Next slide: Guideline 3.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Guideline

Guideline
Slide 4 of 17

◀ ▶

3. Use markup and style sheets and do so properly.

Mark up documents with the proper structural elements. Control presentation with style sheets rather than with presentation elements and attributes.

Using markup improperly -- not according to specification -- hinders accessibility. Misusing markup for a presentation effect (e.g., using a table for layout or a header to change the font size) makes it difficult for users with specialized software to understand the organization of the page or to navigate through it. Furthermore, using presentation markup rather than structural markup to convey structure (e.g., constructing what looks like a table of data with an HTML PRE element) makes it difficult to render a page intelligibly to other devices.

Content developers may be tempted to use (or misuse) constructs that achieve a desired formatting effect on older browsers. They must be aware that these practices cause accessibility problems and must consider whether the formatting effect is so critical as to warrant making the document inaccessible to some users.

At the other extreme, content developers must not sacrifice appropriate markup because a certain browser or assistive technology does not process it correctly. For example, it is appropriate to use the TABLE element in HTML to mark up tabular information. Doing so, and creating tables that transform gracefully (refer to guideline 5), makes it possible for software to render tables other than as two-dimensional grids.

✓ Checkpoints for guideline 3.

Next slide: Guideline 4.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Guideline

4. Clarify natural language usage

Guideline
Slide 5 of 17

◀ ▶

Use markup that facilitates pronunciation or interpretation of abbreviated or foreign text.

When content developers mark up natural language changes in a document, speech synthesizers and braille devices can automatically switch to the new language, making the document more accessible to multilingual users. Content developers should identify the predominant natural language of a document's content (through markup or HTTP headers). Content developers should also provide expansions of abbreviations and acronyms.

In addition to helping assistive technologies, natural language markup allows search engines to find key words and identify documents in a desired language. Natural language markup also improves readability of the Web for all people, including those with learning disabilities, cognitive disabilities, or people who are deaf. Natural language markup may also enable machine translation of documents into other natural languages.

When abbreviations and natural language changes are not identified, they may be indecipherable when machine-spoken or brailled.

✓ Checkpoints for guideline 4.

Next slide: Guideline 5.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Guideline

Guideline
Slide 6 of 17

◀ ▶

5. Create tables that transform gracefully.

Ensure that tables have necessary markup to be transformed by accessible browsers and other user agents.

Tables should be used to mark up truly tabular information ("data tables"). Content developers should avoid using them to lay out pages ("layout tables"). Tables for any use also present special problems to users of screen readers .

Some user agents allow users to navigate among table cells and access header and other table cell information. Unless marked-up properly, these tables will not provide user agents with the appropriate information.

The checkpoints for this guideline will directly benefit people who access a table through auditory means (e.g., a screen reader or an Automobile PC that operates by speech input and output) or who view only a portion of the page at a time (e.g., users with blindness or low vision using speech output or a braille display, or other users of devices with small displays, etc.).

✓ Checkpoints for guideline 5.

Next slide: Guideline 6.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Guideline

Guideline
Slide 7 of 17

◀ ▶

6. Ensure that pages featuring new technologies transform gracefully.

Ensure that pages are accessible even when newer technologies are not supported or are turned off.

Although content developers are encouraged to use new technologies that solve problems raised by existing technologies, they should know how to make their pages still work with older browsers and people who choose to turn off features.

✓ **Checkpoints for guideline 6.**

Next slide: Guideline 7.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Guideline

Guideline
Slide 8 of 17

◀ ▶

7. Ensure user control of time-sensitive content changes.

Ensure that moving, blinking, scrolling, or auto-updating objects or pages may be paused or stopped.

Some people with cognitive or visual disabilities are unable to read moving text quickly enough or at all. Movement can also cause such a distraction that the rest of the page becomes unreadable for people with cognitive disabilities. Screen readers are unable to read moving text. People with physical disabilities might not be able to move quickly or accurately enough to interact with moving objects.

Note. All of the following checkpoints involve some content developer responsibility until user agents provide adequate feature control mechanisms.

Note. The BLINK and MARQUEE elements are not defined in any W3C HTML specification and should not be used. Refer also to guideline 11.

✓ Checkpoints for guideline 7.

Next slide: Guideline 8.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

W3C® Web Accessibility initiative

Chuck Letourneau & Geoff Freed

Guideline

Guideline
Slide 9 of 17

◀ ▶

8. Ensure direct accessibility of embedded user interfaces.

Ensure that the user interface follows principles of accessible design: device-independent access to functionality, keyboard operability, self-voicing, etc.

When an embedded object has its "own interface", the interface - like the interface to the browser itself - must be accessible. If the interface of the embedded object cannot be made accessible, an alternative accessible solution must be provided.

Note. For information about accessible interfaces, please consult the WAI Home Page for pointers to the work being done on User Agent Accessibility Guidelines and Authoring Tool Accessibility Guidelines.

✓ Checkpoints for guideline 8.

Next slide: Guideline 9.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Guideline

Guideline
Slide 10 of 17

◀ ▶

9. Design for device-independence.

Use features that enable activation of page elements via a variety of input devices.

Device-independent access means that the user may interact with the user agent or document with a preferred input (or output) device -- mouse, keyboard, voice, head wand, or other. If, for example, a form control can only be activated with a mouse or other pointing device, someone who is using the page without sight, with voice input, or with a keyboard or who is using some other non-pointing input device will not be able to use the form.

Note. Providing text equivalents for image maps or images used as links makes it possible for users to interact with them without a pointing device.

Generally, pages that allow keyboard interaction are also accessible through speech input or a command line interface.

✓ Checkpoints for guideline 9.

Next slide: Guideline 10.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Guideline

10. Use interim solutions.

Guideline
Slide 11 of 17

◀ ▶

Use interim accessibility solutions so that assistive technologies and older browsers will operate correctly.

For example, older browsers do not allow users to navigate to empty edit boxes. Older screen readers read lists of consecutive links as one link. These active elements are therefore difficult or impossible to access. Also, changing the current window or popping up new windows can be very disorienting to users who have available, but aren't using, the graphical features of the desktop environment.

Note. The following checkpoints apply until user agents and assistive technologies address these issues. These checkpoints are classified as "interim", meaning that the Web Content Accessibility Guidelines Working Group considers them to be valid and necessary to Web accessibility as of the publication of this document. However, the Working Group does not expect these checkpoints to be necessary in the future, once Web technologies have incorporated anticipated features or capabilities.

✓ Checkpoints for guideline 10.

Next slide: Guideline 11.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Guideline

11. Use W3C technologies and guidelines.

Use W3C technologies (according to specification) and follow accessibility guidelines. Where it is not possible to use a W3C technology, or doing so results in material that does not transform gracefully, provide an alternative version of the content that is accessible.

The current guidelines recommend W3C technologies (e.g., HTML, CSS, etc.) for several reasons:

- W3C technologies include "built-in" accessibility features.
- W3C specifications undergo early review to ensure that accessibility issues are considered during the design phase.
- W3C specifications are developed in an open, industry consensus process.

Many non-W3C formats (e.g., PDF, Shockwave, etc.) require viewing with either plug-ins or stand-alone applications. Often, these formats cannot be viewed or navigated with standard Web access or screen reading tools. Avoiding non-W3C and non-standard features (proprietary elements, attributes, properties, and extensions) will tend to make pages more accessible to more people using a wider variety of hardware and software. When proprietary or inaccessible technologies must be used, equivalent accessible pages must be provided.

Even when W3C technologies are used, they must be used in accordance with accessibility guidelines. When using new technologies, ensure that they transform gracefully.

Note. Converting documents (from PDF, PostScript, RTF, etc.) to W3C markup languages (HTML, XML) does not always create an accessible document. Therefore, validate each page for accessibility and usability after the conversion process . If a page does not readily convert, either revise the page until its original representation converts appropriately or provide an HTML or plain text version.

Checkpoints for guideline 11.

Next slide: Guideline 12.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Guideline

12. Provide context and orientation information.

Guideline
Slide 13 of 17

◀ ▶

Provide context and orientation information to help users understand complex pages or elements.

Grouping elements and providing contextual information about the relationships between elements can be useful for all users. Complex relationships between parts of a page may be difficult for people with cognitive disabilities and people with visual disabilities to interpret.

✓ Checkpoints for guideline 12.

Next slide: Guideline 13.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Guideline

Guideline
Slide 14 of 17

◀ ▶

13. Provide clear navigation mechanisms.

Provide clear and consistent navigation mechanisms - orientation information, navigation bars, a site map, etc. - to increase the likelihood that a person will find what they are looking for at a site.

Clear and consistent navigation mechanisms are important to people with cognitive disabilities or blindness, and benefit all users.

✓ **Checkpoints for guideline 13.**

Next slide: Guideline 14.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Guideline

14. Ensure that documents are clear and simple.

Guideline
Slide 15 of 17

◀ ▶

Ensure that documents are clear and simple so they may be more easily understood.

Consistent page layout, recognizable graphics, and easy to understand language benefit all users. In particular, they help people with cognitive disabilities or who have difficulty reading. (However, ensure that images have text equivalents for people who are blind, have low vision, or for any user who cannot or has chosen not to view graphics.)

Using clear and simple language promotes effective communication. Access to written information can be difficult to impossible for people who have cognitive disabilities, learning disabilities, or who are deaf. This consideration also applies to the many people whose first language differs from your own.

✓ Checkpoints for guideline 14.

Next slide: Appendix A.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Guideline

Appendix A. - Validation

Guideline
Slide 16 of 17

◀ ▶

Validate accessibility with automatic tools and human review. Automated methods are generally rapid and convenient but cannot identify all accessibility issues. Human review can help ensure clarity of language and ease of navigation.

Begin using validation methods at the earliest stages of development. Accessibility issues identified early are easier to correct and avoid.

✓ **Methods for Validation.**

Next slide: end of Guideline Set.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

W3C® Web Accessibility initiative

Chuck Letourneau & Geoff Freed

Guideline

End of the Guideline Set

Guideline
Slide 17 of 17

◀

Where to next?

- go to the Introductory Set
- go back to the start of the Guideline Set
- go to the Checkpoint Set
- go to the Example set
- or select from the Table of Contents links below

Any comments or questions?

- about this curriculum? Contact the authors (follow the links at the bottom of this page)
- about the guidelines or techniques? See the WAI Home page for more information.

Introduction | Guidelines | Checkpoints | Examples ◀

Chuck Letourneau & Geoff Freed

Checkpoints for Guideline

Introduction to the Checkpoint Set

Checkpoint
Slide 1 of 17

▶

The following pages are referred to as "The Checkpoint Set" in the introduction to the curriculum.

Each page lists one or more checkpoints that are associated with a particular guideline. For consistency with the published sources, the text of the checkpoints in this set matches that found in the Web Content Accessibility Guidelines V1.0.

For an alternative view of the Web Content Accessibility checkpoints that arranges them according to priority and function, see the appendix that accompanies the Guidelines:
List of Checkpoints for Web Content Accessibility Guidelines 1.0

Each checkpoint links to one or more examples of techniques the author can use to meet the requirements of the checkpoint. Please note that each checkpoint is assigned a priority level. Please read the definitions of the priority levels before proceding. And undestanding of the impact on accessibility of each checkpoint is key to the design process.

Next slide: Checkpoints for Guideline 1

Introduction | Guidelines | Checkpoints | Examples

▶

Chuck Letourneau & Geoff Freed

Checkpoints for Guideline

1. Provide equivalent alternatives to auditory and visual content.

Checkpoint
Slide 2 of 17

Provide content that, when presented to the user, conveys essentially the same function or purpose as auditory or visual content.

1.1 - Provide a text equivalent for every non-text element. Priority 1

- 1.1a - a text equivalent for images and graphical buttons.
- 1.1b - a text equivalent for graphical representations of text (including symbols).
- 1.1c - a text equivalent for image map regions.
- 1.1d - a text equivalent for animations (e.g., animated GIFs).
- 1.1e - a text equivalent for applets and programmatic objects.
- 1.1f - a text equivalent for ASCII art.
- 1.1g - a text equivalent for frames.
- 1.1h - a text equivalent for scripts.
- 1.1i - a text equivalent for images used as list bullets.
- 1.1j - a text equivalent for images used as "spacers".
- 1.1k - a text equivalent for sounds (played with or without user interaction).
- 1.1l - a text equivalent for stand-alone audio files.
- 1.1m - a text equivalent for audio tracks of video.
- 1.1n - a text equivalent for video.

1.2 - Provide redundant text links for each active region of a server-side image map. Priority 1

1.3 - Until user agents can automatically read aloud the text equivalent of a visual track provide an auditory description of the important information of the visual track of a multimedia presentation. Priority 1

1.4 - For any time-based multimedia presentation (e.g., a movie or animation), synchronize equivalent alternatives (e.g., captions or auditory descriptions of the visual track) with the presentation. Priority 1

1.5 - Until user agents render text equivalents for client-side image map links provide redundant text links for each active region of a client-side image map. Priority 3

▲ To Guideline 1

Next slide: Checkpoints for Guideline 2

Introduction | Guidelines | Checkpoints | Examples   ◀ ▶

Chuck Letourneau & Geoff Freed

Checkpoints for Guideline

2. Don't rely on color alone.

Checkpoint
Slide 3 of 17

Ensure that text and graphics are understandable when viewed without color.

2.1 - Ensure that all information conveyed with color is also available without color. Priority 1

2.2 - Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen. Priority 2 / Priority 3

To Guideline 2

Next slide: Checkpoints for Guideline 3

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Checkpoints for Guideline

3. Use markup and style sheets and do so properly.

Checkpoint
Slide 4 of 17

◀ ▶

Mark up documents with the proper structural elements. Control presentation with style sheets rather than with presentation elements and attributes.

3.1 - When an appropriate markup language exists, use markup rather than images to convey information. Priority 2

3.2 - Create documents that validate to published formal grammars. Priority 2

3.3 - Use style sheets to control layout and presentation. Priority 2

3.4 - Use relative rather than absolute units in markup language attribute values and style sheet property values. Priority 2

3.5 - Use header elements to convey document structure and use them according to specification. Priority 2

3.6 - Mark up lists and list items properly. Priority 2

3.7 - Mark up quotations. Do not use quotation markup for formatting effects such as indentation. Priority 2

▲ To Guideline 3

Next slide: Checkpoints for Guideline 4

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Checkpoints for Guideline

4. Clarify natural language usage

Checkpoint
Slide 5 of 17

◀ ▶

Use markup that facilitates pronunciation or interpretation of abbreviated or foreign text.

4.1 - Clearly identify changes in the natural language of a document's text and any text equivalents Priority 1

4.2 - Specify the expansion of each abbreviation or acronym in a document where it first occurs. Priority 3

4.3 - Identify the primary natural language of a document. Priority 3

▲ To Guideline 4

Next slide: Checkpoints for Guideline 5

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Checkpoints for Guideline

5. Create tables that transform gracefully.

Checkpoint
Slide 6 of 17

◀ ▶

Ensure that tables have necessary markup to be transformed by accessible browsers and other user agents.

5.1 - For data tables, identify row and column headers. Priority 1

5.2 - For data tables that have two or more logical levels of row or column headers use markup to associate data cells and header cells. Priority 1

5.3 - Do not use tables for layout unless the table makes sense when linearized. Otherwise, if the table does not make sense, provide an alternative equivalent (which may be a linearized version). Priority 2

5.4 - If a table is used for layout, do not use any structural markup for the purpose of visual formatting. Priority 2

5.5 - Provide summaries for tables. Priority 3

5.6 - Provide abbreviations for header labels. Priority 3

△ To Guideline 5

Next slide: Checkpoints for Guideline 6

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Checkpoints for Guideline

6. Ensure that pages featuring new technologies transform gracefully.

Checkpoint
Slide 7 of 17

◀ ▶

Ensure that pages are accessible even when newer technologies are not supported or are turned off.

6.1 - Organize documents so they may be read without style sheets. When an HTML document is rendered without associated style sheets, it must still be possible to read the document. Priority 1

6.2 - Ensure that equivalents for dynamic content are updated when the dynamic content changes. Priority 1

6.3 - Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page. Priority 1

6.4 - For scripts and applets, ensure that event handlers are input device-independent. Priority 2

6.5 - Ensure that dynamic content is accessible or provide an alternative presentation or page. Priority 2

▲ To Guideline 6

Next slide: Checkpoints for Guideline 7

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Checkpoints for Guideline

7. Ensure user control of time-sensitive content changes.

Checkpoint
Slide 8 of 17

◀ ▶

Ensure that moving, blinking, scrolling, or auto-updating objects or pages may be paused or stopped.

7.1 - Until user agents allow users to control flickering, avoid causing the screen to flicker. Priority 1

7.2 - Until user agents allow users to control blinking, avoid causing content to blink. Priority 2

7.3 - Until user agents allow users to freeze moving content, avoid movement in pages. Priority 2

7.4 - Until user agents provide the ability to stop the refresh do not create periodically auto-refreshing pages. Priority 2

7.5 - Until user agents provide the ability to stop auto-redirect do not use markup to redirect pages automatically. Instead, configure the server to perform redirects. Priority 2

▲ To Guideline 7

Next slide: Checkpoints for Guideline 8

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Checkpoints for Guideline

8. Ensure direct accessibility of embedded user interfaces.

Checkpoint
Slide 9 of 17

◀ ▶

Ensure that the user interface follows principles of accessible design: device-independent access to functionality, keyboard operability, self-voicing, etc.

8.1 - Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies.
Priority 1 / Priority 2

△ To Guideline 8

Next slide: Checkpoints for Guideline 9

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Checkpoints for Guideline

9. Design for device-independence.

Checkpoint
Slide 10 of 17

Use features that enable activation of page elements via a variety of input devices.

9.1 - Provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape. Priority 1

9.2 - Ensure that any element that has its own interface can be operated in a device-independent manner. Priority 2

9.3 - For scripts, specify logical event handlers rather than device-dependent event handlers. Priority 2

9.4 - Create a logical tab order through links, form controls, and objects. Priority 3

9.5 - Provide keyboard shortcuts to important links (including those in client-side image maps), form controls, and groups of form controls. Priority 3

To Guideline 9

Next slide: Checkpoints for Guideline 10

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Checkpoints for Guideline
10. Use interim solutions.

Checkpoint
Slide 11 of 17

◀ ▶

Use interim accessibility solutions so that assistive technologies and older browsers will operate correctly.

10.1 - Until user agents allow users to turn off spawned windows, do not cause pop-ups or other windows to appear and do not change the current window without informing the user. Priority 2

10.2 - Until user agents support explicit associations between labels and form controls, for all form controls with implicitly associated labels, ensure that the label is properly positioned. Priority 2

10.3 - Until user agents render side-by-side text correctly, provide a linear text alternative (on the current page or some other) for all tables that lay out text in parallel, word-wrapped columns. Priority 3

10.4 - Until user agents handle empty controls correctly, include default, place-holding characters in edit boxes and text areas. Priority 3

10.5 - Until user agents render adjacent links distinctly, include non-link, printable characters (surrounded by spaces) between adjacent links. Priority 3

▲ To Guideline 10

Next slide: Checkpoints for Guideline 11

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Checkpoints for Guideline

11. Use W3C technologies and guidelines.

Checkpoint
Slide 12 of 17

◀ ▶

Use W3C technologies (according to specification) and follow accessibility guidelines. Where it is not possible to use a W3C technology, or doing so results in material that does not transform gracefully, provide an alternative version of the content that is accessible.

11.1 - Use W3C technologies when they are available and appropriate for a task and use the latest versions when they are supported. Priority 2

11.2 - Avoid deprecated features of W3C technologies. Priority 2

11.3 - Provide information so that users may receive documents according to their preferences. Priority 3

11.4 - If, after best efforts, you cannot create an accessible page. provide a link to an alternative page that uses W3C technologies, is accessible, has equivalent information (or functionality), and is updated as often as the inaccessible (original) page. Priority 1

▲ To Guideline 11

Next slide: Checkpoints for Guideline 12

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Checkpoints for Guideline
12. Provide context and orientation information.

Checkpoint
Slide 13 of 17

◀ ▶

Provide context and orientation information to help users understand complex pages or elements.

12.1 - Title each frame to facilitate frame identification and navigation. Priority 1

12.2 - Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone. Priority 2

12.3 - Divide large blocks of information into more manageable groups where natural and appropriate. Priority 2

12.4 - Associate labels explicitly with their controls (in a FORM). Priority 2

▲ To Guideline 12

Next slide: Checkpoints for Guideline 13

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Checkpoints for Guideline

13. Provide clear navigation mechanisms.

Checkpoint
Slide 14 of 17

◀ ▶

Provide clear and consistent navigation mechanisms - orientation information, navigation bars, a site map, etc. - to increase the likelihood that a person will find what they are looking for at a site.

13.1 - Clearly identify the target of each link. Priority 2

13.2 - Provide metadata to add semantic information to pages and sites. Priority 2

13.3 - Provide information about the general layout of a site (e.g., a site map, or table of contents). Priority 2

13.4 - Use navigation mechanisms in a consistent manner. Priority 2

13.5 - Provide navigation bars to highlight and give access to the navigation mechanism. Priority 3

13.6 - Group related links, identify the group (for user agents) and, until user agents do so, provide a way to bypass the group. Priority 3

13.7 - If search functions are provided, enable different types of searches for different skill levels and preferences. Priority 3

13.8 - Place distinguishing information at the beginning of headings, paragraphs, lists, etc. Priority 3

13.9 - Provide information about document collections (i.e., documents comprising multiple pages.). Priority 3

13.10 - Provide a means to skip over multi-line ASCII art. Priority 3

▲ To Guideline 13

Next slide: Checkpoints for Guideline 14

Introduction Guidelines Checkpoints Examples ◀ ▶

Chuck Letourneau & Geoff Freed

Checkpoints for Guideline

14. Ensure that documents are clear and simple.

Checkpoint
Slide 15 of 17

◀ ▶

Ensure that documents are clear and simple so they may be more easily understood.

14.1 - Use the clearest and simplest language appropriate for a site's content. Priority 1

14.2 - Supplement text with graphic or auditory presentations where they will facilitate comprehension of the page. Priority 3

14.3 - Create a style of presentation that is consistent across pages. Priority 3

▲ To Guideline 14

Next slide: Checkpoints for Appendix A

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Checkpoints for Guideline

Appendix A. - Validation

Checkpoint
Slide 16 of 17

◀ ▶

Validate accessibility with automatic tools and human review. Automated methods are generally rapid and convenient but cannot identify all accessibility issues. Human review can help ensure clarity of language and ease of navigation.

Validation examples.

△ To Appendix A

Next slide: End of Checkpoint Set

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Checkpoints for Guideline

End of the Checkpoint Set

Checkpoint
Slide 17 of 17

◀

Where to next?

- go to the Introductory Set
- go to the Guideline Set
- go back to the start of the Checkpoint Set
- go to the Example set
- or select from the Table of Contents links below

Any comments or questions?

- About this curriculum? Contact the authors (follow the links at the bottom of this page).
- About the guidelines or techniques? See the WAI Home page for more information.

Introduction | Guidelines | Checkpoints | Examples

◀

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example for Checkpoint

Introduction to the Example Set

Example
Slide 1 of 120

▶

The following pages are referred to as "The Example Set" in the introduction to the curriculum.

Each page provides details or examples of one or more techniques that are associated with a particular checkpoint. Where possible, the examples are actually coded so that you will see how that particular technique displays or renders on your browser or user agent. In most cases, the markup that creates the "live" example is also provided (although you can also "View Source" to get the exact coding). Where necessary, some text is included to explain what is "supposed" to happen (for example, if an HTML 4 attribute is not widely supported yet), or for user of non-graphical or older browsers.

In this set, the examples of techniques are drawn from various sources: some from the authors' experience, some from other W3C sources, and some from the document "Techniques for Web Content Accessibility Guidelines 1.0". Please note that the "Techniques" document is the definitive source

For an alternative view of the Web Content Accessibility checkpoints that arranges them according to priority and function, see the appendix that accompanies the Guidelines:
List of Checkpoints for Web Content Accessibility Guidelines 1.0

Next slide: Example for Checkpoint 1.1a.

Introduction | Guidelines | Checkpoints | Examples

▶

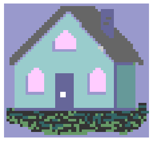
Chuck Letourneau & Geoff Freed

Example for Checkpoint

1.1a - a text equivalent for images and graphical buttons.

Example
Slide 2 of 120

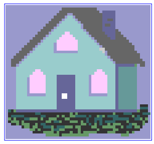
Priority 1



For a simple, decorative or incidental image, a brief description may be all that is necessary: e.g.

This drawing of a house is just a drawing of a house, so the code could be:

<IMG SRC="home.gif" ALT="**Drawing of a house**.">



If the image is linked, you must describe the destination or purpose of the link -- not the image, e.g.

Here, the drawing of a house is a button that links to the home page, so the code could be:

```
<A HREF="home.htm">
<IMG SRC="home.gif" ALT="Link to the Home page.">
</A>
```

To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.1a continues

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint 1.1a, continued.

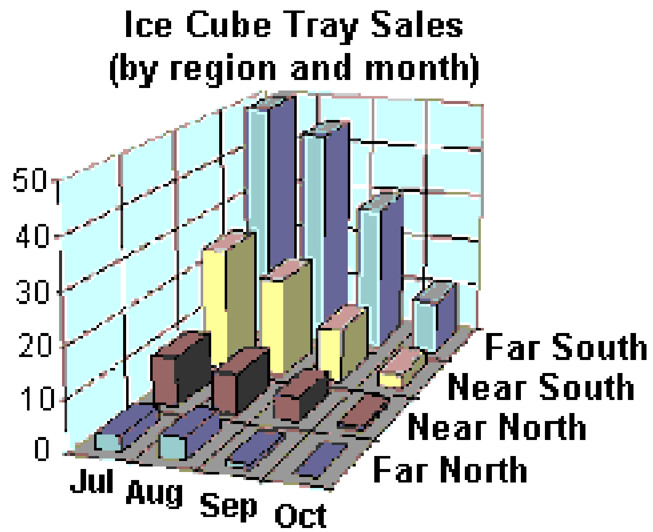
Example
Slide 3 of 120

Priority 1

If the information contained in the image is important to the meaning of your page (i.e. some important content would be lost if the image was removed) , then you must provide a longer description than the "alt" attribute can reasonably display. The "longdesc" attribute was created for this reason.

<IMG SRC="graph1.gif" LONGDESC="graph1.htm" ALT="3-d sales chart.">

Future browsers or other agents will provide an optional a link to the description file called "graph1.htm".



To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.1a continues

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

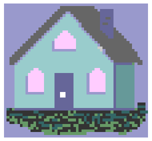
Example for Checkpoint 1.1a, continued.

Example
Slide 4 of 120

◀ ▶

Priority 1

If you use the OBJECT element to place an image,



then include the text alternative in the content of the OBJECT element, e.g.

<OBJECT data="home.gif" type="image/gif" height=100 width=100>
Place the short or long description (or link to a long description file) here.
</OBJECT>

△ To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.1b

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example
Slide 5 of 120

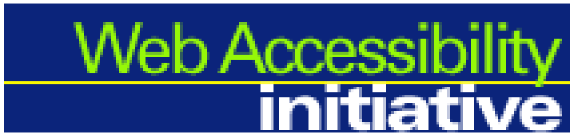
◀ ▶

Example for Checkpoint

1.1b - a text equivalent for graphical representations of text (including symbols).

Priority 1

For an image that is simply a bit-map of text (because you want to use special graphical font-effects or other transformations that would be difficult or impossible using style sheets), provide the text equivalent, e.g.



<IMG SRC="wai.gif" ALT="Web Accessibility Initiative">

Where a special markup language for symbols exists - such as MathML - use that instead of a simple text equivalent. See the example for Checkpoint 3.1 for more details.

△ To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.1c

Introduction Guidelines Checkpoints Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

1.1c - a text equivalent for image map regions.

Example
Slide 6 of 120

◀ ▶

Priority 1: for server side image maps

For the user of a non-graphical browser or for someone who does not use a traditional pointing device (like a mouse or touch-screen), a page that uses an imagemap as the sole means of navigation can be quite inaccessible. There are a number of techniques available to you to ensure that your image maps are accessible. All of those techniques involve providing a text equivalent for the critical functions of the map: namely the active areas that activate links to other information.

See the example for Checkpoint 1.2 for a more detailed treatment of using a server side image map .

Priority 3: for client side image maps

See the example for Checkpoint 1.5 for more information on using a client side image map .

▲ To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.1d

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 7 of 120

1.1d - a text equivalent for animations (e.g., animated GIFs).

Priority 1

For example,

```
<IMG SRC="ani-bal.gif" ALT="An animated expanding and bursting balloon.">
```

To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.1e

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 8 of 120

1.1e - a text equivalent for applets and programmatic objects.

Priority 1

APPLET is deprecated in HTML 4.0 (which means you should start using the new OBJECT element instead), but it is still in frequent use. If you are modifying an APPLET to make it more accessible, use the "alt" attribute to briefly describe the action or purpose. Also, if it will help the audience to understand what the APPLET is meant to conveys, include a more detailed description in the content of the APPLET element (i.e. between the <APPLET> and </APPLET> tags), e.g.

If you were using a Java-enabled browser, you would see the text "It was the best of times, it was the worst of times..." dancing across the screen, instead of this paragraph.

If your browser supports APPLET, and if you have opted to let Java applets run on your system, then you will see some text randomly changing color and appearing in the APPLET window. If either condition is not met, you should see the following text:

If you were using a Java-enabled browser, you would see the text "It was the best of times, it was the worst of times..." dancing across the screen, instead of this paragraph.

Select this link to see the code that created the dancing text and its alternative.

To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.1f

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

1.1f - a text equivalent for ASCII art.

Example
Slide 9 of 120

Priority 1: if the ASCII art conveys important information.

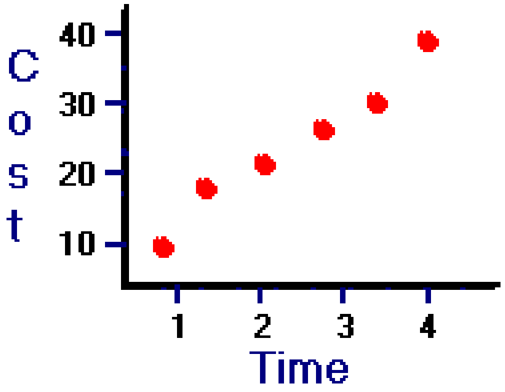
For our report on a manufacturing process, we have created two charts showing the relationship of cost versus time. The chart made with ASCII characters will be meaningless to users of screen readers. The second chart shows the same information in a bit-map image created using simple drawing software. While the image is inaccessible to users of screen-readers, at least we can add equivalent text using the alt or longdesc attributes. (See the example for Checkpoint 1.1a for an example of using "longdesc"

```
ASCII-art graph

 C  40-|         o
 o  30-|     o o
 s  20-|  o o
 t  20-| o
        ------------
         '  '  '  '

         1  2  3  4
            Time
```

Bit-mapped image graph



To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.1f continues

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint 1.1f, continued.

Example
Slide 10 of 120

◀ ▶

Priority 2: if the ASCII art is not critical to the content.

If the ASCII art image does not convey critical information, a simple description is recommended, e.g.,
The following ASCII art picture is a bearded person wearing a hat. (Skip over ASCII art)

```
     MMMMMMMM
 WWWWW      WWWWW
******************
   | \O/ \O/ |
   |    |    |
   \         /
    \  ---  /
      VVVVV
       VVV
```

A bearded person wearing a hat - drawn with ASCII characters.

Note the link to allow a person using a screen-reader to jump over the ASCII art and quickly read the caption.

▲ To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.1g

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

1.1g - a text equivalent for frames.

Example
Slide 11 of 120

◀ ▶

Priority 1

Use the "title" and, if necessary, the "longdesc" attributes of the FRAME element to describe the frame or its relationship with other frames.

Please go to the example for Checkpoint 12.2 for more details.

△ To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.1h

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example for Checkpoint

1.1h - a text equivalent for scripts.

Example
Slide 12 of 120

◀ ▶

Priority 1

Use the NOSCRIPT element to describe the action or replace the functionality of any scripts you embed in your page.

For example, if you write a SCRIPT that animates a comic strip by presenting a new panel whenever the user mouse-clicks or presses a key, then use the NOSCRIPT element to provide a text alternative that describe the entire comic strip. For example:

```
<SCRIPT type="text/javascript">
... script to display and replace each frame of the comic strip as requested by a user event ...
</SCRIPT>

<NOSCRIPT>
In the first panel, Lucy is holding the football. In the second panel, Charlie Brown is seen charging towards the football. And so on..."
</NOSCRIPT>
```

▲ To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.1i

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example
Slide 13 of 120

Example for Checkpoint
1.1i - a text equivalent for images used as list bullets.

Priority 1

Here is a simple unordered list that uses graphical bullets instead of the default bullets created by the <UL> element:

● Pencil
● Telephone
● Calendar

The example code shows three different but appropriate ways to use the alt parameter to label the bullet.

```
<IMG SRC="blueball.gif" ALT="Item: "> Pencil <BR>
<IMG SRC="blueball.gif" ALT="* "> Telephone <BR>
<IMG SRC="blueball.gif" ALT=" "> Calendar <BR>
```

- You might use "Item" or some other simple text to highlight the item, especially if there are not many entries in the list.
- You might use an asterisk "*" or a dash "-" to provide a common text alternative to a graphical bullet.
- If you have a large number of items in your list, then it might be appropriate to type a blank space between the quotation marks of the alt attribute. This would save the user of a screen reader from hearing "Item..." repeated time after time.

To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.1j

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

1.1j - a text equivalent for images used as "spacers".

Example
Slide 14 of 120

◀ ▶

Priority 1

A layout trick used by many authors is to create a very small transparent image file and force white space between other objects by setting the required height and width or hspace and vspace attributes of the IMG element. The appropriate text equivalent for a "spacer" would be nothing at all, so:

```
<IMG src="spacer.gif" hspace=100 vspace=10 alt=" ">
```

Note that there is a blank space between the quotation marks in the alt-attribute. Of course, we would prefer you use style sheet markup to do your layout.

△ To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.1k

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 15 of 120

1.1k - a text equivalent for sounds (played with or without user interaction).

Priority 1

You are a clever programmer. You have written a script that causes a warning sound, like a "Oh-oh!", to be played if the visitor to your page tries to submit a form before all the required fields have been completed.

Submit

If you are that clever, please give your program (or script) the ability to write a message to the screen that says something like:

"You have tried to submit an incomplete form. Please complete the required fields."

Access note: This example does not use a script or a form. We simulated what could be done with a script using simple HTML.

To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.1l

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

1.1l - a text equivalent for stand-alone audio files.

Example
Slide 16 of 120
◀ ▶

Priority 1

If the sound file is very short, you could include the transcript in the alt-text of an image that accompanies the link to the sound file, e.g.

Listen to a message from the author.

```
<A HREF="work-e.wav">
<IMG SRC="audio.gif" ALT="Sound file: Let's work together for accessibility. ">
Listen to a message from the author. </A>
```

If the audio file contains lots of information, then you might want to link to a file that contains a complete transcript:

Let's listen to an excerpt from "The Walrus and the Carpenter," by Lewis Carroll.

Audio transcript: Excerpt from "The Walrus and the Carpenter," by Lewis Carroll.

Man:
"The time has come," the Walrus said,
"To talk of many things:
Of shoes-- and ships-- and sealing-wax--
Of cabbages-- and kings--
And why the sea is boiling hot--
And whether pigs have wings."

▲ To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.1m

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

1.1m - a text equivalent for audio tracks of video.

Example
Slide 17 of 120

◀ ▶

Priority 1

A text equivalent for the audio track of a video would be similar to the textual transcript of the spoken voices recorded in an audio file. This can be either a synchronized text equivalent (as described in Checkpoint 1.3) - the preferred method, or a simple text file, as shown here:

Man:
Go long, way out.

Professor:
Submitted for your consideration: the ball has been thrown, and you want to catch it. But how do you know where it will go so you can be there when it comes down?

One way is to use this formula.

It factors in velocity, acceleration and time to calculate the distance the ball will go. But then, you knew that. Your brain estimates all of these values in the first seconds of the ball's flight to calculate where you have to go to catch the ball.

This math equation helps us understand the physical world and how we function in it. Whether you do it on the board or on the field, they both work. So, next time you go out for that long pass, remember: Math is everywhere.

Math is everywhere!

See QuickTime instructions for the movie clip on which this transcript is based.

▲ To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.1n

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

1.1n - a text equivalent for video.

Example
Slide 18 of 120

◀ ▶

Priority 1

A text equivalent for video would be a description of the scenes and actions that are taking place in the video - but not the spoken dialog. In this example, the description of what is happening in the video is interspersed with the text equivalent of the dialog. As in the previous example, the preferred method would be to integrate and synchronize the textual equivalent of the video in the movie clip itself.

Select this link for an example of a transcript of a QuickTime movie that includes complete captioning and video description.

See QuickTime instructions for the movie clip on which this transcript is based.

▲ To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.2

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 19 of 120
◀ ▶

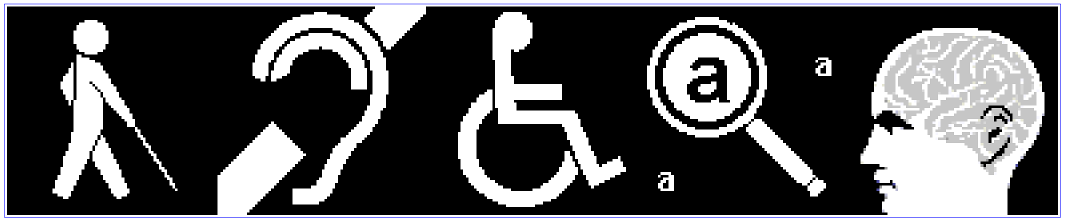
1.2 - Provide redundant text links for each active region of a server-side image map.

Priority 1

Server-side image maps (those using the ISMAP attribute in the IMG element) usually don't or can't provide any textual information about the links that are coded into them. If your server-side image map has hot-links to sections A, B, C, D and E of your site, then provide a text alternative on the page. For example, the code:

```
<A HREF="img/imgmap1.map">
<IMG ISMAP SRC="imgmap1.gif"
ALT="Please use the following links instead of this imagemap.">
</A><BR>
[ <A HREF="a.htm">Section A</A> | <A HREF="b.htm">Section B</A> | <A HREF="c.htm">Section C</A> | <A HREF="d.htm">Section D</A> | <A HREF="e.htm">Section E</A> ]
```

produces the following server-side image map and text equivalent:



[Section A | Section B | Section C | Section D | Section E]

Note: the "alt" text you provide in the IMG element informs the users that a text equivalent exists -- but does not describe the image itself. If you want to describe the image map in detail, use "longdesc".

▲ To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.3

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 20 of 120

◀ ▶

1.3 - Until user agents can automatically read aloud the text equivalent of a visual track, provide an auditory description of the important information of the visual track of a multimedia presentation.

Priority 1

An auditory description of a multimedia clip's visual track is used to convey important information, such as scenery, movement, charts, graphs, etc., which is otherwise lost if the user cannot see the screen. For persons who are blind or visually impaired, these audio descriptions are necessary if the user is to fully understand any video-based presentation.

This guideline is only a stop-gap measure because we expect that future user agents will be able to use a text-to-speech application to automatically read and speak audio descriptions from text. Writing the descriptions (rather than recording them) may be easier for page authors since only a text editor is required. Providing recorded audio descriptions requires the page author to have sound recording and editing capabilities. Another benefit of written descriptions (as opposed to recorded descriptions) is that the text file can be indexed and searched, just like any other text on a Web site.

▲ To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.3 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 1.3 - continued

Example
Slide 21 of 120
◀ ▶

However, since there are currently no user agents that can automatically read and speak written descriptions, a multimedia presentation must be accompanied by an audio track which describes important information presented in the visual track. The next example (Checkpoint 1.4) explains the need to synchronize all the equivalent alternatives of a multimedia presentation.

At present there are three formats or languages that support the addition of a separate audio description track. These are Apple's QuickTime, the W3C's SMIL (Synchronized Multimedia Integration Language) and Microsoft's SAMI. You will need different software to play the different formats, so check the multimedia software page to download what's appropriate. From the multimedia software page, we provide links to examples that demonstrate the auditory description track.

▲ To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.4

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example
Slide 22 of 120

◀ ▶

Example for Checkpoint

1.4 - For any time-based multimedia presentation, synchronize equivalent alternatives with the presentation.

Priority 1

A time-based presentation can include any form of multimedia, such as a movie, animation or slide show. Equivalent alternatives to these types of presentations are captions (which provide access to audio tracks) and audio descriptions (which provide access to visual tracks).

We have already explained the need to provide a textual transcript for any audio track (see Checkpoint example 1.1l) or video track (see Checkpoint example 1.1m), and a textual description of the video track (see Checkpoint examples 1.1n and 1.3). However, it must be admitted that a text transcript alone is not the ideal method for providing an equitable experience for persons with disabilities. It is widely accepted that on-screen captioning allows deaf and hard-of-hearing people to more fully appreciate the experience of a movie or multimedia production. An almost-equivalent accommodation for people with sight and hearing is the provision of subtitles during foreign-language films or performances. A separate textual transcript that must be read after the fact does not provide an equivalent experience.

Thus the requirement to synchronize the equivalent alternatives. The caption track is an alternative for deaf or hearing-impaired viewers. The audio-description track is an alternative for people who are blind or visually impaired. Synchronizing these alternatives with the main presentation (that is, the video and/or audio) means that nearly all users will get the best experience and the most information available to them. (Bear in mind that for people who do not have access to multimedia-playback devices, or for people who are deaf-blind, a transcript of both the audio and the audio descriptions is still the best alternative.)

▲ To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.4 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 1.4 - continued

Example
Slide 23 of 120

◀ ▶

How do you ensure that your movies or multimedia meet this guideline?

1. In many cases multimedia projects are created for you by professional studios which already have (or ought to have) the ability to produce the accessible equivalents. When you contract for this type of work, include the accessible alternatives in your core requirements.

2. If you produce your own multimedia or movie clips, it is likely that you already have the tools on hand to build in your own accessible equivalents. Many standard sound, movie, animation packages (including those from Apple's QuickTime or RealNetworks) can be used to create accessible formats.

At present there are three formats or languages that support the ability to synchronize equivalent alternatives. These are Apple's QuickTime, the W3C's SMIL (Synchronized Multimedia Integration Language) and Microsoft's SAMI. You will need different software to play the different formats, so check the multimedia software page to download what's appropriate. From the multimedia software page, we provide links to examples that demonstrate the synchronization of equivalent alternatives.

△ To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 1.5

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example for Checkpoint

1.5 - Until user agents render text equivalents for client-side image map links, provide redundant text links for each active region of a client-side image map.

Priority 3

At the present time, it may not be enough to provide 'alt' text describing the destination of the link in the AREA element that creates the geometric hot-spots on an imagemap. To ensure everyone can navigate your site, it is still necessary to provide redundant text links.

For a client-side image map describe the destination that each active area will link to. For example:

[Section A | Section B | Section C | Section D]

View the markup that created this example of a client side image map and its text alternative.

In the future, most new browsers will be able to display the alternative text provided with the AREA element.

▲ To Checkpoints for Guideline 1.

Next slide: Example for Checkpoint 2.1

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example for Checkpoint

Example
Slide 25 of 120

◀ ▶

2.1 - Ensure that all information conveyed with color is also available without color, for example from context or markup.

Priority 1

☺ The first example uses color to highlight the preferred choice for those who can perceive it, and an ordered list to identify the preferred choice for people who cannot see colors.

Example 1: There are two ways to get things done around here. The second method, also shown in red, is preferred.

1. Your way.
2. My way.

☹ Don't use color to convey information unless the information is also clear from the markup and/or text. Without color, the following example is meaningless. (This example is intentionally colorless to show the author that if the person visiting their page cannot see color, they will miss the point the author is trying to make.)

Example 2: There are two ways to get things done around here. The preferred method is shown in red.

- Your way.
- My way.

▲ To Checkpoints for Guideline 2.

Next slide: Example for Checkpoint 2.2

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

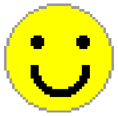
Example for Checkpoint

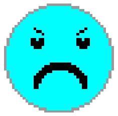
Example
Slide 26 of 120

2.2 - Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen.

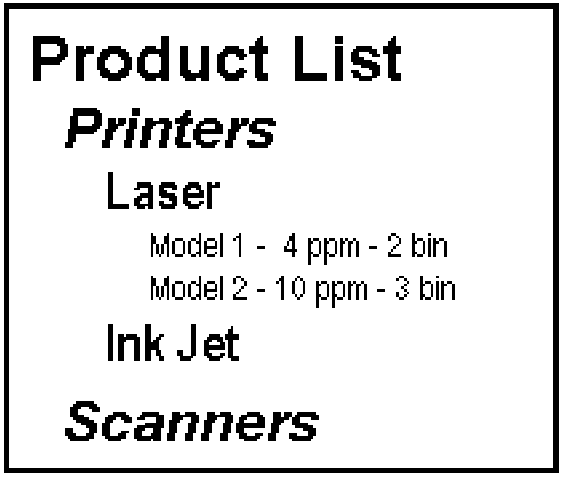
Priority 2: for images.

The first example shows a bit-mapped image containing dark text on a light background, which is good contrast for viewing in almost any circumstance. The second example shows an image with poor choices for foreground and background color. Some people or some devices might not easily interpret the content. The reason the priority is higher for color choices in an image is that the user has no control over the rendering of the color of an image.



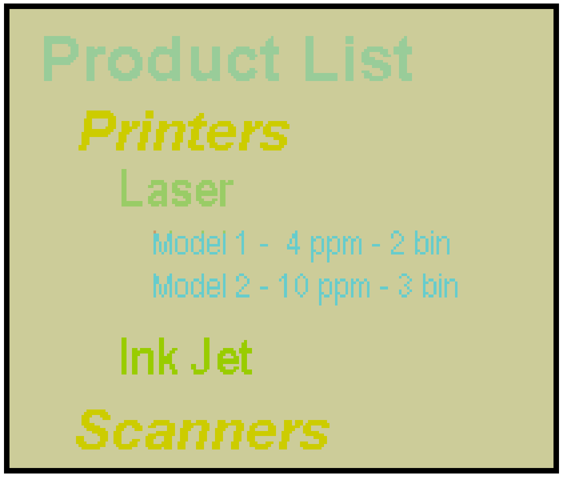


Example 1



Product List
Printers
Laser
Model 1 - 4 ppm - 2 bin
Model 2 - 10 ppm - 3 bin
Ink Jet
Scanners

Example 2



Product List
Printers
Laser
Model 1 - 4 ppm - 2 bin
Model 2 - 10 ppm - 3 bin
Ink Jet
Scanners

Access note: the actual wording contained in these example images is the same as in the following examples of text color.

△ To Checkpoints for Guideline 2.

Next slide: Example for Checkpoint 2.2 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example for Checkpoint 2.2, continued.

Example
Slide 27 of 120

◀ ▶

Priority 3: for text.

The reason that safe color choices are a lower priority for text is that most (if not all) browsers that support color let you chose your own foreground and background colors, overriding the page author's choices. In the case of style sheet markup, browsers will let you select your own preferred stylesheet as well. Thus, in general you have some control over font colors. Still, if your page has poor color choices, many visitors will be inconvenienced.

Access note: These examples are intended to show people who can see color the difference between good and bad color choices - Text example 1 shows text with good contrast (black text on white background), while Text example 2 shows bad contrast (light yellow, green and blue text on a greenish-brown background. This example also uses CSS style to format and color the text, so if your browser doesn't support CSS you will see only plain text in both examples. Finally, a two column table is used to position these examples because CSS style sheet positioning doesn't work well across all browsers yet.

☺	☹
Text color - Example 1	Text color - Example 2
Product List	Product List
Printers	Printers
Laser	Laser
Model 1 - 4 ppm - 2 bin	Model 1 - 4 ppm - 2 bin
Model 2 - 10 ppm - 3 bin	Model 2 - 10 ppm - 3 bin
Ink Jet	Ink Jet
Scanners	Scanners

▲ To Checkpoints for Guideline 2.

Next slide: Example for Checkpoint 3.1

Introduction Guidelines Checkpoints Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

3.1 - When an appropriate markup language exists, use markup rather than images to convey information.

For example, use MathML to mark up mathematical equations, and style sheets to format text and control layout. Also, avoid using images to represent text -- use text and style sheets instead.

Priority 2

Mathematical and scientific equations and symbols and musical notations are often presented as bit-mapped images, like the one in this example. The image of the equation is inaccessible, and alt-text is usually insufficient to convey the meaning of the equation. Groups like the W3C are developing markup languages that will ease the creation of and permit the accessible interpretation of special notations.

One particular language being developed is called MathML.
Select this link to see the MathML markup that might generate the following equation,

$$\int_0^t \frac{dx}{x}$$

▲ To Checkpoints for Guideline 3.

Next slide: Example for Checkpoint 3.2

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example for Checkpoint

Example
Slide 29 of 120

3.2 - Create documents that validate to published formal grammars.

Priority 2

In general the overall accessibility of any Web page will be improved if you adhere to the standards for the web-presentation language you are using. The major reason for this is that most developers of assistive technology or accessible web technology have based their accessibility features on the premise that content authors are following the standards.

We also assume that most Web browsers will more or less follow the published standards. Writing the "cleanest" code possible should ensure your pages will look best across the widest range of Web display technology.

If your coding meets the requirements of a particular grammar, use the DOCTYPE statement as the first line of your HTML file. For example, the following DOCTYPE statement would indicate to servers, browsers and validators that you expect your code to conform to the HTML 4.0 Transitional document type descriptor (or DTD). The DTD is the "formal" grammar of the language.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

Quite a few sites, including the W3C, offer HTML validation. There are even sites that will validate your HTML and check your page for some aspects of accessibility. See the section on Validation and Testing in this curriculum for more details.

To Checkpoints for Guideline 3.

Next slide: Example for Checkpoint 3.3

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

3.3 - Use style sheets to control layout and presentation.

Example
Slide 30 of 120

◀ ▶

Priority 2

Use style sheets to control text color and sizing. Elements such as BASEFONT, and FONT size, color etc. are deprecated in HTML 4.0 so it doesn't hurt to start weaning yourself off them now. As we also told you, please don't use the heading elements (H1-H6) just to get quick font size changes. The following example shows some style sheet commands that will do the job very well:

It isn't the cough that carries you off,
it's the coffin they carry you off in.

```
<STYLE TYPE="text/css">
.para { font-size: 130%; color: #ff0000 }
.linetwo { font-size: 70% ; font-style: italic ; color: #0000ff }
</STYLE>

<P CLASS=para>It isn't the cough that carries you off,
<SPAN CLASS=linetwo>it's the coffin they carry you off in.</SPAN></P>
```

▲ To Checkpoints for Guideline 3.

Next slide: Example for Checkpoint 3.3 continues

Introduction Guidelines Checkpoints Examples ◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 3.3, continued.

Example
Slide 31 of 120

◀ ▶

Use style sheet commands rather than BLOCKQUOTE, UL, DL & DT, TABLE and so on just to get a quick formatting boost. Each of those HTML elements have their proper use in defining the structure of a document. For all those simple formatting tricks you did using HTML, please use CSS instead. For example, you can create a nice colored background for an indented paragraph without creating a TABLE:

If your browser supports style sheets, you will see a block of text with a red border, indented from the margins of the page, and sporting a light blue background.

And the style command to create this is simply:

<DIV style="border: thin red groove; padding: 1em; margin: 2%; background: #ccffff">If your browser...</DIV>

△ To Checkpoints for Guideline 3.

Next slide: Example for Checkpoint 3.4

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example
Slide 32 of 120

◀ ▶

Example for Checkpoint

3.4 - Use relative rather than absolute units in markup language attribute values and style sheet property values.

Priority 2

For example, in CSS, use 'em' or percentage lengths rather than 'pt' or 'cm', which are absolute units. If absolute units are used, validate that the rendered content is usable by checking the results on various browsers or systems. See the section on Validation and Testing for more on that subject.

▲ To Checkpoints for Guideline 3.

Next slide: Example for Checkpoint 3.5

Introduction Guidelines Checkpoints Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example for Checkpoint

3.5 - Use header elements to convey logical structure and use them according to specification.

Example
Slide 33 of 120

◀ ▶

Priority 2

Nest headings in order, as in this example. Some user agents will use the heading structure to improve or add useful navigation features, or to abstract the document. Using heading levels appropriately will ensure your document is more accessible.

Page title: use (<H1>)

First major section heading: use (<H2>)

First sub-section title: use (<H3>)

This is where the content of the sub section will go. It may be paragraphs, lists, quotes, diagrams or whatever your document requires.

Second sub-section title: use (<H3>)

Second major section heading: use (<H2>)

First sub-section title: use (<H3>)

▲ To Checkpoints for Guideline 3.

Next slide: Example for Checkpoint 3.5 continues

Introduction Guidelines Checkpoints Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 3.5, continued

Example
Slide 34 of 120
◀ ▶

Don't nest headings randomly (as in this next example) just to take advantage of the default display sizes of the H elements. Instead, use CSS to create your visual text effects.

Incorrect use of <H5>: We will never offer

Incorrect use of <H1>: Free Software Upgrades

Incorrect use of <H1>: Because we care

Incorrect use of <H5>: only about our bottom line.

To Checkpoints for Guideline 3.

Next slide: Example for Checkpoint 3.6

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

3.6 - Mark up lists and list items properly.

Priority 2

Encode list structure and list items (UL, OL, DL, LI) properly. The HTML list elements DL, UL, and OL (available in HTML 3.2 and HTML 4.0) should only be used to create lists, not for formatting effects such as indentation. When possible, use ordered (numbered) lists to help navigation.

Structure the list with navigational information, like this:

. Instruments found within the string section of a symphony orchestra:

    1. Violins
    2. Violas
    3. Cellos

And the code for this example is:

```
<OL type=A>
<LI>Instruments found within the string section of a symphony orchestra:</LI>
<OL type=1>
<LI>Violins</LI>
<LI>Violas</LI>
<LI>Cellos</LI>
</OL>
```

To Checkpoints for Guideline 3.

Next slide: Example for Checkpoint 3.6 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 3.6, continued.

Example
Slide 36 of 120

◀ ▶

Priority 2

☹ Don't organize a list in an unstructured manner, like this:

- Instruments found within the string section of a symphony orchestra:

  Violins

  Violas

  Cellos

```
<UL>
<LI>Instruments found within the string section of a symphony orchestra:
<DL>
<DT>Violins
<DT>Violas
<DT>Cellos
</DL></UL>
```

This list was created with the DL and DT elements, which should be used for definition lists, not simply for indenting a list. Instead, use the unordered list element <UL> if this is really a list, or use CSS if all you want is the indentation provided by <DL>, for example:

```
<P style="text-indent: 3em">Violins</P>
```

will indent an item without adding any unwanted semantics to the markup.

△ To Checkpoints for Guideline 3.

Next slide: Example for Checkpoint 3.7

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example
Slide 37 of 120

◀ ▶

Example for Checkpoint

3.7 - Mark up quotations. Do not use quotation markup for formatting effects such as indentation.

Priority 2

Mark up quotations with the Q and BLOCKQUOTE elements. Do not use them for formatting effects such as indentation.

<Q> element:
Quotation marks and quotation styles vary from language to language. Using the Q element (instead of typing the ASCII equivalent of quotation marks) will allow international browsers to properly display the required quote system. Therefore, please use the Q element to insert quotation marks, like this:

"I'm going home at midnight!"

<Q lang="en-us"> I'm going home at midnight!</Q>

Access Note: The Q element is not widely supported yet. Until it is, you will have to continue to add quotation marks manually. The "lang" attribute of the Q element is expected to cause the language-specific quotation symbols or rules to be applied during the rendering of the quote. To the best of our knowledge, this feature is also not supported by any browser.

▲ To Checkpoints for Guideline 3.

Next slide: Example for Checkpoint 3.7 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 3.7, continued.

Example
Slide 38 of 120

◀ ▶

<BLOCKQUOTE> element:

The BLOCKQUOTE element is meant to identify a block of text as a quote - separate and distinct from the surrounding text. That is why most browsers indent text identified with the BLOCKQUOTE tags. However, because most browsers did this people began using BLOCKQUOTE to do some simple page layout. We encourage you to return BLOCKQUOTE to its original purpose and use stylesheets to do your text formatting. Use the BLOCKQUOTE element only to offset quoted blocks of text in a meaningful manner, like this:

In chapter 7, Bob speaks his most important words:

> I'm going home at midnight!

The BLOCKQUOTE tags surround the actual quotation in this example:

In chapter 7, Bob speaks his most important words:
<BLOCKQUOTE> I'm going home at midnight!</BLOCKQUOTE>

▲ To Checkpoints for Guideline 3.

Next slide: Example for Checkpoint 3.7 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example for Checkpoint 3.7, continued.

Example
Slide 39 of 120

◀ ▶

Note: Your browser must support CSS in order for you to see this example inset from the margins of the page

Use stylesheet markup if you just want to indent a block of text.

This simple style markup in the P element causes the example to be indented from the left and right margins:

```
<P style="margin-left: 10%; margin-right: 10%; >Bob was going home...</P>
```

Bob was going home at midnight, because the overnight train was less crowded than the rush-hour special. He was worried about falling asleep and missing his stop, but at least he would have a seat to himself.

Or, in a declared style section,

```
P.inset {margin-left: 10%; margin-right: 10%;}
```

then in the HTML:

```
<P class="inset">Bob was going home...</P>
```

▲ To Checkpoints for Guideline 3.

Next slide: Example for Checkpoint 4.1

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 40 of 120
◀ ▶

4.1 - Clearly identify changes in the natural language of a document's text and any text equivalents

Priority 1

Use the "lang" attribute to clearly identify changes in the language of text.

When mixing languages in a sentence or paragraph, notate the language changes like this:

Mother, he's asking you to go.
He's saying, "Allons, Madame plaisante!"

And the code would be:

Mother, he's asking you to go.
He's saying,<SPAN lang="fr"> "Allons, Madame plaisante!"</SPAN>

For a complete list of language codes, check out:
Code for the Representation of the Names of Languages. From ISO 639, revised 1989. Source: OASIS - the Organization for the Advancement of Structured Information Standards.

▲ To Checkpoints for Guideline 4.

Next slide: Example for Checkpoint 4.2

Introduction | Guidelines | Checkpoints | Examples ◀ ▶

Chuck Letourneau & Geoff Freed

Example
Slide 41 of 120

Example for Checkpoint

4.2 - Specify the expansion of each abbreviation or acronym in a document where it first occurs.

Priority 3

When using abbreviations or acronyms in a sentence, be sure to indicate this in the HTML using ABBR and ACRONYM, like this:

> When in Boston, be sure to visit the MFA, MIT and, of course, the W3C. These destinations are easily reached via Mass. Ave. or Mem. Dr..

And the markup would look like this:

```
When in Boston, be sure to visit the
<ACRONYM TITLE="Museum of Fine Arts">MFA</ACRONYM>,
<ACRONYM TITLE="Massachusetts Institute of Technology">MIT</ACRONYM>
and, of course, the
<ACRONYM TITLE="World Wide Web Consortium">W3C</ACRONYM>.
These destinations are easily reached via
<ABBR TITLE="Massachusetts Avenue">Mass. Ave.</ABBR>
or <ABBR TITLE="Memorial Drive">Mem. Dr.</ABBR>.
```

Access Note: in the near future, browsers will let you choose to have the abbreviations and acronyms expanded automatically.

Or, provide the expansion (especially of the first occurrence) in the main body of the document (i.e. the way you would if you were not using HTML).

To Checkpoints for Guideline 4.

Next slide: Example for Checkpoint 4.3

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 42 of 120

4.3 - Identify the primary natural language of a document.

Priority 3

The language of the entire document should be specified in the HTML tag. In this example, the primary language of the content is Italian.

```
<HTML lang="it">

<HEAD></HEAD>

<BODY>

<P>Volete sapere l'origine della parola cappuccino,

il popolare caffelatte all'italiana?</P>

</BODY>

</HTML>
```

In XML, use "xml:lang".

For a complete list of language codes, check out:
Code for the Representation of the Names of Languages. From ISO 639, revised 1989. Source: OASIS - the Organization for the Advancement of Structured Information Standards.

For information about content negotiation, see the example for Checkpoint 11.3.

To Checkpoints for Guideline 4.

Next slide: Example for Checkpoint 5.1

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

5.1 - For data tables, identify row and column headers.

Example
Slide 43 of 120

Priority 1

This checkpoint is telling you that if you have a legitimate need to present data in a tabular format, then you should use the HTML TABLE element and its supporting elements and attributes (like TR, TD, TH and CAPTION). The alternatives (such as using the PRE tag for preformatted text or using style sheets) will actually make understanding of tabular data more difficult for users of assistive technology.

A simple data table created with proper data table mark up, might look like this:

Example of a simple data table created using HTML markup.

	Column 1 header	**Column 2 header**
Row 1 header	Column 1 Row 1	Column 2 Row 1
Row 2 header	Column 1 Row 2	Column 2 Row 2

See the code that created this table.

The above example is preferable to the using the <PRE> element to layout the data, because future browsers will use the TH and other TABLE markup to logically linearize tables.

Note: if you are using tables for page layout (instead of CSS), then you should NOT use markup reserved for data tables (like TH, HEADER, SCOPE, COLGROUP, etc.) because those elements will be used by some agents to identify and manipulate data).

To Checkpoints for Guideline 5.

Next slide: Example for Checkpoint 5.1 continues

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint 5.1, continued.

Example
Slide 44 of 120

HTML 4.0 also allows you to explicitly link header information to columns and rows using the "headers" attribute of the <TD> and <TH> elements, e.g.:

Cups of coffee consumed by each senator

Name	Cups	Type of Coffee	Sugar?
T. Sexton	10	Espresso	No
J. Dinnen	5	Decaf	Yes

If you use the 'headers' attribute, a browser or screen reader might be able to expose or read the contents of the cells (if the user wishes) like this:

Name: T. Sexton, Cups: 10, Type: Espresso, Sugar: No
Name: J. Dinnen, Cups: 5, Type: Decaf, Sugar: Yes

because each datum is explicitly associated with its appropriate header.

View the markup code that would generate this example.

To Checkpoints for Guideline 5.

Next slide: Example for Checkpoint 5.2

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example
Slide 45 of 120

◀ ▶

Example for Checkpoint

5.2 - For data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells.

Priority 1

For "complex" tables, i.e. where tables have structural divisions beyond those implicit in the rows and columns, use appropriate markup to identify those divisions.

Example: A travel expenses worksheet. While the following data table appears simple enough visually, it would be difficult to understand if read by some of today's screen-readers. A good way to approximate what some screen-reader users will hear is to hold a ruler to the table, and read straight across the screen. Then, move the ruler down until the next line of characters is displayed. Read straight across. Acter a while, pick a data cell at random and, without looking at the column or row title, try and remember what headers describe that data point. The larger and more complex the table, the harder it would be to remember the row and column relationships.

Example 1: TRAVEL EXPENSES (actual cost, US$)

TRIP, date	**Meals**	**Room**	**Trans.**	**Total**
San Jose				
25 Aug 97	37.74	112.00	45.00	
26 Aug 97	27.28	112.00	45.00	
Subtotal	65.02	224.00	90.00	379.02
Seattle				
27 Aug 97	96.25	109.00	36.00	
28 Aug 97	35.00	109.00	36.00	

Subtotal	131.25	218.00	72.00	421.25
Totals	196.27	442.00	162.00	800.27

To Checkpoints for Guideline 5.

Next slide: Example for Checkpoint 5.2 continues

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint 5.2, continued.

Example
Slide 46 of 120

◀ ▶

Today, most users of screen-readers would hear this table read as:

TRAVEL EXPENSES (actual cost, US$)
TRIP,
Meals Room Trans Total
date
San Jose
25 Aug 97 37.74 112.00 45.00
26 Aug 97 27.28 112.00 45.00
Subtotal 65.02 224.00 90.00 379.02
Seattle
27 Aug 97 96.25 109.00 36.00
28 Aug 97 35.00 109.00 36.00
Subtotal 131.25 218.00 72.00 421.25
Totals 196.27 442.00 162.00 800.27

Try reading this out loud to yourself and when you get to the 11th line, try to guess what the meaning of the fourth value is supposed to be... without looking back at the header information.

▲ To Checkpoints for Guideline 5.

Next slide: Example for Checkpoint 5.2 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example for Checkpoint 5.2, continued.

Example
Slide 47 of 120

◀ ▶

Tomorrow's smart browsers or screen-readers will use additional HTML 4.0 markup (TBODY, THEAD, SCOPE, HEADERS, etc.) to "intelligently" decode a table for speech or alternative output. This table might be read as follows:

TRAVEL EXPENSES (actual cost, US$)
Trip: San Jose, Date: 25 Aug 97, Meals: 37.74, Room: 112.00, Trans. 45.00
Trip: San Jose, Date: 26 Aug 97, Meals: 27.28, Room: 112.00, Trans. 45.00
Trip: San Jose, Subtotal, Meals: 65.02, Room: 224.00, Trans. 90.00, Total: 379.02
Trip: Seattle, Date: 27 Aug 97, Meals: 96.25, Room: 109.00, Trans. 36.00
Trip: Seattle, Date: 28 Aug 97, Meals: 35.00 Room: 109.00, Trans. 36.00
Trip: Seattle, Subtotal, Meals: 131.25, Room: 218.00, Trans. 72.00, Total: 421.25
Trip: Totals: Meals: 196.27, Room: 442.00, Trans: 162.00, Total: 800.27

View an example of the TABLE markup that would render the table both visually and provide the extra information needed to understand the information in different presentations.

▲ To Checkpoints for Guideline 5.

Next slide: Example for Checkpoint 5.3

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

5.3 - Do not use tables for layout unless the table makes sense when linearized.

Example
Slide 48 of 120

Otherwise, if the table does not make sense, provide an alternative equivalent (which may be a linearized version).

Priority 2

As more browsers include consistent support for the W3C Cascading Style Sheet Recommendations CSS 1and CSS 2, it will become increasingly proper to do all positioning of text and images on a page using style sheet markup. TABLE should only be used for the proper markup of data that needs to be presented in tabular form. This checkpoint is really just one more example of the requirement to use markup for its intended function, not for the layout "features" it appears to provide. See the examples of checkpoints for Guideline 3 for examples of style commands used for layout.

To Checkpoints for Guideline 5.

Next slide: Example for Checkpoint 5.4

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

5.4 - If a table is used for layout, do not use any structural markup for the purpose of visual formatting.

Priority 2

You already know we would prefer you to use Cascading Style Sheet language to lay out and format the text and images on your page. We also know that TABLE is still used widely for positioning text and image block on Web pages. If you feel you must use TABLE to lay out text elements, then please use CSS rather than TABLE or other HTML markup to style your text.

For example, use:

<TD><DIV style="font-weight: bold">Give me bolded text</DIV></TD>

rather than:

<TH>Give me bolded text<TH>

TH is in HTML to mean "Table Header", not to give you another way to bold some text.

To Checkpoints for Guideline 5.

Next slide: Example for Checkpoint 5.5

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example for Checkpoint

5.5 - Provide summaries for tables.

Example
Slide 50 of 120

◀ ▶

Priority 3

Cups of coffee consumed by each senator

Name	Cups	Type of Coffee	Sugar?
T. Sexton	10	Espresso	No
J. Dinnen	5	Decaf	Yes

Hidden in the <TABLE> element of this example is the "summary" attribute:

```
<TABLE border="1" summary="This table charts the number of
cups of coffee consumed by each senator,
the type of coffee (decaf or regular),
and whether taken with sugar.">
... table markup ...
</TABLE>
```

A browser or screen reader will be able to expose or read the contents of the summary attribute if the user wishes.

△ To Checkpoints for Guideline 5.

Next slide: Example for Checkpoint 5.6

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example for Checkpoint

5.6 - Provide abbreviations for header labels.

Example
Slide 51 of 120

Priority 3

A useful feature of <TH> element is the "abbr" or abbreviation attribute. A screen-reader will be able to speak the header identified with each cell if the id and header, or scope, attributes are used. If you don't use the "abbr" attribute, the entire header will be spoken for its associated data values. For example, the code:

<TH id="t3" abbr="Type">Type of Coffee</TH>

would cause a screen-reader to say "Type" and the cell value, instead of "Type of coffee:" and the cell value.

Here is where you would really appreciate the use of "abbr": Imagine if you had a large spreadsheet whose columns were labeled like this:

<TH id="t4" abbr="Projected">Projected spending on coffee in the Western Regional Office</TH>

How many times do you need to hear "Projected spending on coffee in the Western Regional Office" before you wish you had abbreviated it to "Projected"?

To Checkpoints for Guideline 5.

Next slide: Example for Checkpoint 6.1

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example
Slide 52 of 120

◀ ▶

Example for Checkpoint

6.1 - Organize documents so they may be read without style sheets. When an HTML document is rendered without associated style sheets, it must still be possible to read the document.

Priority 1

Style markup is used to create the following fancy text display:

the lazy dog.
The quick
jumped over
brown fox

Let's say you use style commands that do not specify the vertical positioning of each part of the sentence, for example:

```
<STYLE TYPE="text/css">

.part1 /* The quick */ { padding-left: 0;

color: red; font-size: 14pt;

font-family: copperplate gothic bold, fantasy, sans-serif }

.part2 /* brown fox */ {padding-left: 100px;

color: brown; font-size: 10pt;

font-family: times new roman, desdemona, serif }
```

```
.part3 /* jumped over */ { padding-left: 350px;

color: purple; font-size: 18pt;

font-family: desdemona, times new roman, serif }

.part4 /* the lazy dog */ { padding-left: 350px;

color: blue; font-size: 24pt;

font-family: fantasy, copperplate gothic bold, sans-serif }

</STYLE>
```

Using the above style commands, you would have to order the text like this in your HTML:

```
<DIV class=part4>the lazy dog.</DIV>
<DIV class=part1>The quick</DIV>
<DIV class=part3>jumped over</DIV>
<DIV class=part2>brown fox</DIVv>
```

So, if style sheets were turned off or not supported in your browser, you would read:

> the lazy dog
> The quick
> jumped over
> brown fox

△ To Checkpoints for Guideline 6.

Next slide: Example for Checkpoint 6.1 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 6.1, continued.

Example
Slide 53 of 120

◀ ▶

This styled text appears to be the same as in the previous example:

The quick
brown fox
jumps over
the lazy dog

This example uses style attributes to specify the vertical as well as the horizontal positioning:

```
<STYLE TYPE="text/css">

<!--

.part1 /* The quick */ { color: red; font-size: 14pt;

padding-left: 0; margin-top: 40px;

font-family: copperplate gothic bold, fantasy, sans-serif }

.part2 /* brown fox */ { color: brown; font-size: 10pt;

padding-left: 100px; margin-top: 30px;

font-family: times new roman, desdemona, serif }

.part3 /* jumped over */ { color: purple; font-size: 18pt;

padding-left: 200px; margin-top: -60px;
font-family: desdemona, times new roman, serif }

.part4 /* the lazy dog */ { color: blue; font-size: 24pt;

padding-left: 350px;

margin-top: -100px; margin-bottom: 100px;

font-family: fantasy, copperplate gothic bold, sans-serif }

-->

</STYLE>
```

By letting the style commands lay out your text on the display, you can order the content logically in the HTML file, like this:

```
<DIV class=part1>The quick</DIV>

<DIV class=part2>brown fox</DIV>

<DIV class=part3>jumped over</DIV>

<DIV class=part4>the lazy dog.</DIV>
```

So, if style sheets were turned off or not supported in your browser, you would read:

> The quick
> brown fox
> jumped over
> the lazy dog

▲ To Checkpoints for Guideline 6.

Next slide: Example for Checkpoint 6.2

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 54 of 120

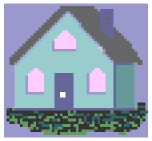
◀ ▶

6.2 - Ensure that equivalents for dynamic content are updated when the dynamic content changes.

Priority 1

Always make the source of a frame an HTML document

Access note: This simple example uses a table and an image to simulate the look of a page created with FRAME markup.

	menu	The accessible way to insert content into a frame is to make the frame's source an HTML file. In this case, source for the frame with the picture of the house should be: ☺ <FRAME name="picture_frame" src="home.html"> where "home.html" should contain the following: <IMG src="home.gif" alt="Picture of a house."> . ☹ If you make the frame's source the image file itself: <FRAME name="picture_frame" src="home.gif"> then you can't provide a text equivalent for the image.

▲ To Checkpoints for Guideline 6.

Next slide: Example for Checkpoint 6.2 continues

Introduction Guidelines Checkpoints Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C Web Accessibility initiative

Example for Checkpoint 6.2, continued.

Example
Slide 55 of 120

Access note: This simple example uses a table and an animated image to simulate the look of a dynamic page created with FRAME markup.

	menu	You might think it would be appropriate to use the title attribute of the FRAME element to label or describe the content of the frame, but the difficulty arises if you are doing dynamic content changes. If you change the source you cannot easily also change the title on the FRAME so the content would change while the title remained constant. If however, the source of each new image or other content is an HTML file, that file can contain all the text equivalents or explanatory information the user needs to understand the changes. ☹ In this example, the picture frame might be created with the following code: `<FRAME src="image-changer.html" title="picture of a house">` However, the title, 'picture of a house', would not be an appropriate title for the frame since the image changes from a house to a telephone and back. ☺ A more appropriate title would be 'changing picture frame'. This begs the question: how can you provide meaningful alt-text for random images dynamically served to a page? Can anyone suggest a suitable technique? Contact the authors with your suggestions.

To Checkpoints for Guideline 6.

Next slide: Example for Checkpoint 6.3

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example
Slide 56 of 120

Example for Checkpoint

6.3 - Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page.

Priority 1

Content developers must ensure that pages are accessible with scripts turned off or in browsers that don't support scripts.

- Avoid creating content on the fly locally (client-side). If a user's browser does not handle scripts, no content will be generated or displayed. However, this is different than displaying or hiding already existing content by using a combination of style sheets and scripting; if there is no script, then the content is always shown. This also does not rule out generating pages on the fly on the server-side and delivering them to the client.

- Avoid creating links that use "javascript" as the URI. If a user is not using scripts, then they won't be able to link since the browser can't create the link content. For example, do NOT do this:

  ```
  <A href="javascript:">...</A>
  ```

  Because this is a dead-end link for a user agent where scripts are not supported or not loaded.

To Checkpoints for Guideline 6.

Next slide: Example for Checkpoint 6.4

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

6.4 - For scripts and applets, ensure that event handlers are input device-independent.

Example
Slide 57 of 120

◀ ▶

Priority 2

You Win First Prize!

To ensure people get the equivalent message on systems that don't support scripts and event-handlers, include some equivalent content in the <NOSCRIPT> element. As well ensure that the event-handlers you use are logical rather than device-dependent (See Checkpoint 9.3 for more about using event-handlers accessibly.)

The following example displays a BUTTON. The button is associated with a JavaScript SCRIPT. When the script is triggered by an EVENT, an ALERT window pops up on the screen of a graphical browser. The alert window contains a message for the user. In this case, the event is either the TAB key moving the focus to the button, or the user passing the mouse cursor over the button. The 'OnFocus' EVENT-HANDLER looks for the first type of event, while the 'OnMouseOver' event-handler looks for the second.

Win a prize!

The script is followed by the NOSCRIPT element that contains the same message that would have appeared in the alert window. If BUTTON and SCRIPT (or the particular script-language) are not supported, a browser should display the content of BUTTON and the NOSCRIPT elements. In that case, you should see:

Win a prize!
You Win First Prize!

To see the markup for this example, please follow this link.

▲ To Checkpoints for Guideline 6.

Next slide: Example for Checkpoint 6.5

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

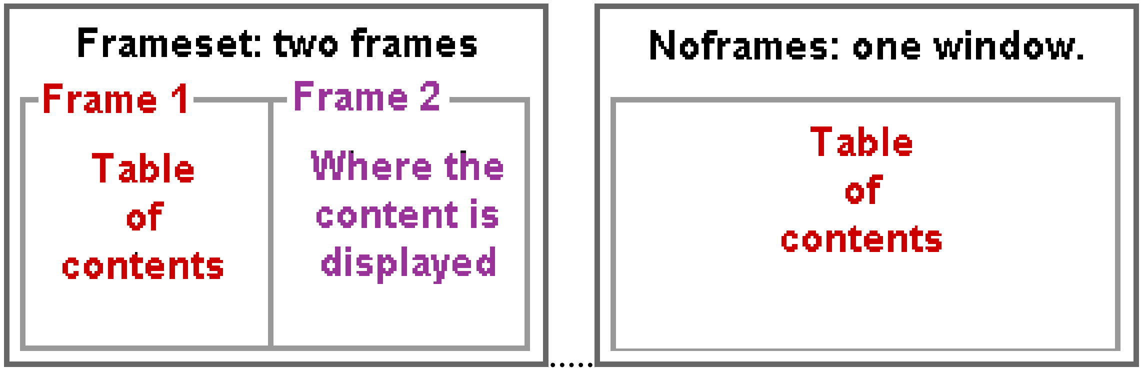
Example
Slide 58 of 120

Example for Checkpoint

6.5 - Ensure that dynamic content is accessible or provide an alternative presentation or page.

Priority 2

Example 1: in HTML, use NOFRAMES at the end of each FRAMESET.



See the code that could generate both the frame and noframe versions of a page.

To Checkpoints for Guideline 6.

Next slide: Example for Checkpoint 6.5 continues

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint 6.5 - continued.

Example
Slide 59 of 120

◀ ▶

Example 2: in HTML, use NOSCRIPT with SCRIPT.

The following code would run a Tcl script in browsers that support that script language, while the content of the NOSCRIPT element would be displayed in browsers that do not support that script language.

```
<SCRIPT type="text/tcl">
...some Tcl script to show a billboard of sports scores...
</SCRIPT>
<NOSCRIPT>
<P>Results from yesterday's games:</P>
<DL>
<DT>Bulls 91, Sonics 80.
<DD><A href="bullsonic.html">Bulls vs. Sonics game highlights</A>
...more scores... </DL>
</NOSCRIPT>
```

Note: For some applications, server-side scripts may be more accessible than client-side scripts.

▲ To Checkpoints for Guideline 6.

Next slide: Example for Checkpoint 7.1

Introduction Guidelines Checkpoints Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 60 of 120

◀ ▶

7.1 - Until user agents allow users to control it, avoid causing the screen to flicker.

Priority 1

People with photosensitive epilepsy can have seizures triggered by flickering or flashing in the 4 to 59 flashes per second (Hertz) range with a peak sensitivity at 20 flashes per second as well as quick changes from dark to light (like strobe lights).

Of course, we can't show you an example of an image or text blinking at 20 flashes per second or we could be accused of trying to bring on an seizure.

The point is: be careful when designing animated .gif images or other refreshing content.

▲ To Checkpoints for Guideline 7.

Next slide: Example for Checkpoint 7.2

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example for Checkpoint

Example
Slide 61 of 120

7.2 - Until user agents allow users to control it, avoid causing content to blink.

Priority 2

If blinking content (e.g., a headline that appears and disappears at regular intervals) is used, provide a mechanism for stopping the blinking.

If you feel you must highlight some text by causing it to *flash* on and off, then use the CSS, 'text-decoration: blink' attribute. Using CSS will allow users to stop the effect by turning off style sheets or overriding the rule in a user style sheet.

Do not use the BLINK and MARQUEE elements. These elements are not part of any W3C specification for HTML (i.e., they are non-standard elements).

To Checkpoints for Guideline 7.

Next slide: Example for Checkpoint 7.3

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 62 of 120

◀ ▶

7.3 - Until user agents allow users to freeze moving content, avoid movement in pages.

Priority 2

When a page includes moving content provide a mechanism within a script or applet to allow users to freeze motion or updates. Using style sheets with scripting to create movement allows users to turn off or override the effect more easily.

▲ To Checkpoints for Guideline 7.

Next slide: Example for Checkpoint 7.4

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example
Slide 63 of 120

Example for Checkpoint

7.4 - Until user agents provide the ability to stop the refresh do not create periodically auto-refreshing pages.

Priority 2

If your page is being continually updated, inform the user that they should reload the page often (rather than by doing it for them).

For example, in HTML, don't cause pages to auto-refresh with "HTTP-EQUIV=refresh" until user agents allow users to turn off the feature. The following code would cause the same page to "refresh" or reload itself every 60 seconds. Presumably you would do this because you are changing the content of that page frequently.

```
<META http-equiv="refresh" content="60">
<BODY>
<P>...Information...
</BODY>
```

To Checkpoints for Guideline 7.

Next slide: Example for Checkpoint 7.5

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 64 of 120

7.5 - Until user agents provide the ability to stop auto-redirect do not use markup to redirect pages automatically. Instead, configure the server to perform redirects.

Priority 2

The following code might automatically bring up the new page five seconds after you load the current page.

```
<HEAD>
<TITLE>Don't use this!</TITLE>
<META http-equiv="refresh" content="5;
http://www.acme.com/newpage">
</HEAD>
<BODY>
<P>If your browser supports Refresh,
you'll be transported to our <A href="http://www.acme.com/newpage">new site</A> in 5 seconds, otherwise, select the link
manually.
</BODY>
```

However, authors should not redirect users with this markup since is non-standard, it disorients users, and it can disrupt a browser's history of visited pages. Instead, in order of preference, authors should:

1. Configure the server to use the appropriate HTTP status code (301). Using HTTP headers is preferable because it reduces Internet traffic and download times, it may by applied to non-HTML documents, and it may be used by agents who requested only a HEAD request (e.g., link checkers). Also, status codes of the 30x type provide information such as "moved permanently" or "moved temporarily" that cannot be given with META refresh.
2. Replace the page that would be redirected with a static page containing a normal link to the new page.

To Checkpoints for Guideline 7.

Next slide: Example for Checkpoint 8.1

Introduction Guidelines Checkpoints Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example
Slide 65 of 120

◀ ▶

Example for Checkpoint

8.1 - Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies.

Priority 1: if functionality is important and not presented elsewhere, and...

Priority 2: otherwise.

If an applet (created with either OBJECT or APPLET) requires user interaction (e.g., the ability to manipulate a physics experiment) that cannot be duplicated in an alternative format, make the applet directly accessible.

The accessibility of objects with their own interface is independent of the accessibility of the user agent. Accessibility must therefore be built into the objects or an alternative must be provided. If you are a programmer you should be aware of the resources available to help you ensure your programs are accessible.

For more information about accessible applets, please see:
Note: these links take you away from this presentation.

- Java Accessibility -- Trace R&D Center
- IBM Guidelines for Writing Accessible Applications Using 100% Pure Java -- IBM Special Needs Systems
- Active Accessibility, Microsoft
- Advanced Microsoft Visual Basic 6.0, 2nd Edition, introduces developers to the use of VB 6.0 in writing applications for Active Accessibility

△ To Checkpoints for Guideline 8.

Next slide: Example for Checkpoint 9.1

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example
Slide 66 of 120

Example for Checkpoint

9.1 - Provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape.

Priority 1

The purpose of this checkpoint, in the context of device-independence, is to remind us that client-side image maps can (and must) use the alt attributes on the AREA elements to let the user know the purpose (destination) of each active area. Server side image maps cannot, in most circumstances, provide the user with similar information. If you must use a server side image map, you must provide an alternative way for users to navigate the links hidden in the image map.

Server side image maps are still important for applications like geographical information systems and mapping applications where each point (coordinate pair) is active. Designing a client side map with thousands of single pixel active regions would be very difficult.

To Checkpoints for Guideline 9.

Next slide: Example for Checkpoint 9.1 continues

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint 9.1, continued.

Another method of creating a client side image map uses the <OBJECT>, <MAP> and <A> elements (although this method is not widely supported yet):

Navigate the site.
[Section A | Section B | Section C | Section D]

This code should expose the standard links if the user's browser does not support OBJECT or has turned off the loading of images:

```
<OBJECT data="imgmap4.gif" type="image/gif"

usemap="#map1">

<MAP name="map1">

<P>Navigate the site. </P>

<A href="a.htm" shape="rect"

coords="0,0,39,39">[Section A]</A>

<A href="b.htm" shape="rect"

coords="40,0,79,39">[Section B]</A>

<A href="c.htm" shape="rect"

coords="80,0,120,39">[Section C]</A>

<A href="d.htm" shape="rect"

coords="121,0,160,39">[Section D]</A>

</MAP>

</OBJECT>
```

See Checkpoint 1.2 for an example using a server side image map
and Checkpoint 1.5 for an example with a client side image map.

▲ To Checkpoints for Guideline 9.

Next slide: Example for Checkpoint 9.2

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example
Slide 68 of 120

Example for Checkpoint

9.2 - Ensure that any element that has its own interface can be operated in a device-independent manner.

Priority 2

Not every user has a graphic environment with a mouse or other pointing device. Some users rely on keyboard, alternative keyboard or voice input to navigate links, activate form controls, etc. Content developers should always ensure that users may interact with a page with devices other than a pointing device. A page designed for keyboard access (in addition to mouse access) will generally be accessible to users with other input devices. What's more, designing a page for keyboard access will usually improve its overall design as well.

Image map links:

> Provide text equivalents for client-side image map areas, or provide redundant text links for server-side image maps. Refer to the image map section for examples.

To Checkpoints for Guideline 9.

Next slide: Example for Checkpoint 9.2 continues

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint 9.2, continued.

Example
Slide 69 of 120

◀ ▶

Keyboard access to links and form controls may be specified in a few ways:

Keyboard shortcuts:

> Keyboard shortcuts allow users to combine keystrokes to navigate links or form controls on a page. Note. Keyboard shortcuts may be handled differently by different operating systems, the primary difference being which keys to press to activate the shortcut. On Windows machines, the "alt" and "ctrl" key are most commonly used while on a Macintosh, it is the apple or "clover leaf" key.

Tabbing order:

> Tabbing order describes a (logical) order for navigating from link to link or form control to form control (usually by pressing the "tab" key, hence the name).

Device-independent control for embedded interfaces:

> Some elements import objects whose interfaces cannot be controlled through the markup language. For example, in HTML: applets, and multi-media players. In such cases, users should ensure that if the imported objects themselves do not provide accessible interfaces, that an alternative does.

▲ To Checkpoints for Guideline 9.

Next slide: Example for Checkpoint 9.3

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example
Slide 70 of 120

Example for Checkpoint

9.3 - For scripts, specify logical event handlers rather than device-dependent event handlers.

Priority 2

An event-handler invokes a script when a certain event occurs (e.g, the mouse moves, a key is pressed, the document is loaded, etc.). In HTML 4.0, event handlers are attached to elements via event handler attributes (the attributes beginning with "on", as in "onkeyup").

What happens when an event occurs depends on the script the page author has created. Some produce purely decorative effects such as highlighting an image or changing the color of an element's text. Others produce much more substantial effects, such as carrying out a calculation, providing important information to the user, or submitting a form. For scripts that do more than just change the presentation of an element, content developers should do the following:

Use application-level event triggers rather than user interaction-level triggers. In HTML 4.0, application-level event attributes are "onfocus", "onblur" (the opposite of "onfocus"), and "onselect". Note that these attributes are designed to be device-independent, but are implemented as keyboard specific events in current browsers.

Otherwise, if you must use device-dependent attributes, provide redundant input mechanisms (i.e., specify two handlers for the same element):

- Use "onmousedown" with "onkeydown".
- Use "onmouseup" with "onkeyup"
- Use "onclick" with "onkeypress"
- Note that there is no keyboard equivalent to double-clicking ("ondblclick") in HTML 4.0.

To Checkpoints for Guideline 9.

Next slide: Example for Checkpoint 9.3 continues

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

First Prize!

Example for Checkpoint 9.3, continued.

Example
Slide 71 of 120

◀ ▶

The following examples can only be fully appreciated if your browser supports JavaScript (and you have enabled scripts to be run). Your browser must also support the BUTTON element.

Example 1: Use a device independent event handler that works with any appropriate input action: in this case, ONFOCUS works with both a keyboard event and a mouse event, so it is considered to be a device independent event handler.

The example is the "script activated button": try clicking on it or tabbing to it. A small alert box should pop up containing a message about the prize. Pressing the Escape key will make it go away.

Win a prize!

And here is the code that creates it:

```
<BUTTON onfocus="pop1()"> Win a prize!</BUTTON>.
```

And here is the script that does the work:

```
<SCRIPT language="JavaScript" type="text/javascript">
function pop1() { alert ("First Prize!") }
</SCRIPT>
<NOSCRIPT>First Prize!</NOSCRIPT>
```

▲ To Checkpoints for Guideline 9.

Next slide: Example for Checkpoint 9.3 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 9.3, continued.

Example
Slide 72 of 120

◀ ▶

Second Prize!

Example 2: Use more than one device-dependent event handler to "cover all the bases": ONCLICK requires a mouse event, while ONKEYPRESS requires a keyboard event. Both are device dependent, but including both should allow anyone to benefit from the technique.

Another "script-activated button": click on it, or tab to it and press any key to activate the script. A small alert box should pop up containing a message about the prize. Pressing the Escape key will make it go away.

Win a different prize!

Here is the code that creates the button:

```
You will win a small prize if you
<BUTTON ONCLICK="pop2()" ONKEYPRESS="pop2()">
Win a different prize!</BUTTON>.
```

And here is the code for the script function:

```
<SCRIPT language="JavaScript" type="text/javascript">
function pop2() { alert ("Second Prize!") }
</SCRIPT>
<NOSCRIPT>Second Prize!</NOSCRIPT>
```

Finally, do not write event handlers that rely on mouse coordinates since this prevents device-independent input.

▲ To Checkpoints for Guideline 9.

Next slide: Example for Checkpoint 9.4

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

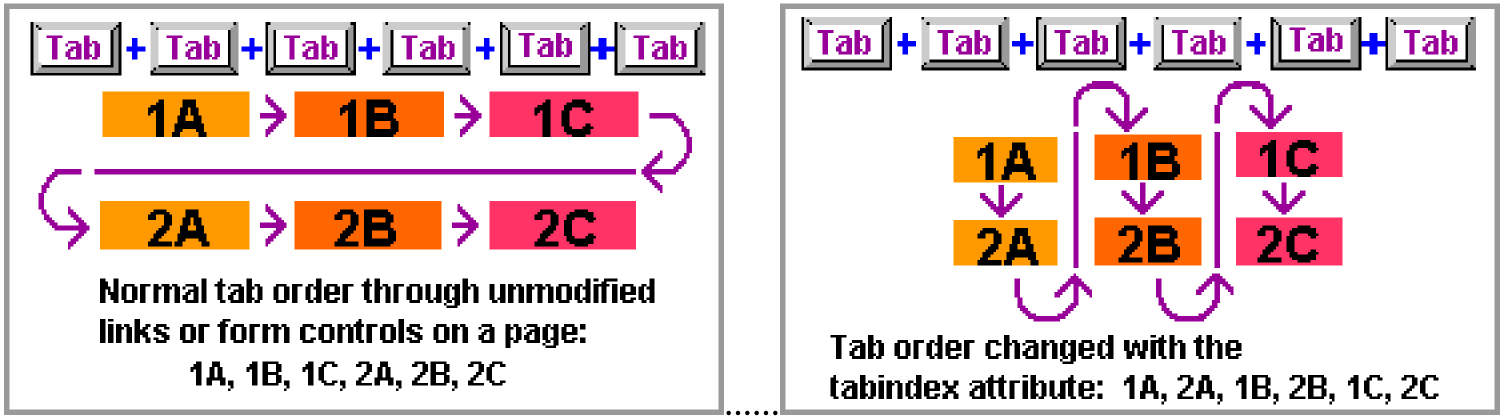
Example for Checkpoint

9.4 - Create a logical tab order through links, form controls, and objects.

Example
Slide 73 of 120

◀ ▶

Priority 3


Tab + Tab + Tab + Tab + Tab + Tab
1A
1B
1C
2A
2B
2C
Tab + Tab + Tab + Tab + Tab + Tab
1A
1B
1C
2A
2B
2C

Normal tab order through unmodified links or form controls on a page: 1A, 1B, 1C, 2A, 2B, 2C

Tab order changed with the tabindex attribute: 1A, 2A, 1B, 2B, 1C, 2C

In browsers that have enabled the Tab key to navigate between links and form controls, the default tabbing order starts with the first link or control on the page, and then jumps to each subsequent link or control in the order it appears in the HTML markup for the page. This default order may not always be convenient, but it can be changed to suit your needs using the tabindex attribute. Tabindex works with the <A> <AREA> <BUTTON>, <INPUT>, <OBJECT>, <SELECT> and <TEXTAREA> elements.

To Checkpoints for Guideline 9.

Next slide: Example for Checkpoint 9.4 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 9.4 - continued.

Example
Slide 74 of 120

a. Using TABINDEX:

The lay out of fields on the following form example is intentionally confusing. However, because the TABINDEX code specifies a logical order for the fields, using the TAB key should present them to you in the "correct" order (for browsers that support the tabkey attribute - in fact, this specific example may only work (or work as the author expects) if you are using Internet Explorer 4.0 or later):

Note: please do **NOT** try to submit this example form. It may cause an error on some systems.

Submit

(6th tab stop)

What's your favorite color?

Red□

(3rd tab stop)

Green□

(4th tab stop)

Blue□

(5th tab stop)

Company Name

(2nd tab stop)

Reset

(7th tab stop)

Skip the form to find accessible alternatives.

(1st tab stop)

Select this link to see the markup that created this example.

▲ To Checkpoints for Guideline 9.

Next slide: Example for Checkpoint 9.4 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 9.4, continued.

Example
Slide 75 of 120

Of course, you could just design the form more logically so that tabbing would lead logically through the fields or links:

Note: please do **NOT** try to submit this example form. It may cause an error on some systems.

Skip the form to find accessible alternatives.

Company Name ▭

What's your favorite color?
Red□
Green□
Blue□

Enter Undo

To Checkpoints for Guideline 9.

Next slide: Example for Checkpoint 9.5

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 76 of 120

◀ ▶

9.5 - Provide keyboard shortcuts to important links (including those in client-side image maps), form controls, and groups of form controls.

Priority 3

HTML 4.0 makes it possible to "keyboard enable" anchors and form controls. For instance, the following form form has been keyboard enabled for browsers that support the accesskey attribute. In this example, the active keys are:

- a - to skip the form to find accessible alternatives;
- s - to submit the form,
- u - to undo the form,
- m - to enter company name,
- r, g, or b - to select the color red, green or blue as your favorite color.

In the example below, the active keys are also underlined or, on input buttons, separated from the rest of the word and capitalized. Please note that this highlighting was done manually... the accesskey attribute does not cause any highlighting or otherwise identify the chosen letter in any known browser.

Also, please note that browsers that support the accesskey attribute may do so differently. If you use Internet Explorer 4.0 or 5.0, you can hold down the ALT-key and press the corresponding letter to see how that browser supports the accesskey attribute (IE 4.0 and 5.0 differ in how they support accesskey). Other browsers may use other attention-getting keys, or not support accesskey at all.

Note: please do **NOT** try to submit this example form. It may cause an error on some systems.

Skip the form to find **A**ccessible alternatives.

[S submit] [U ndo]

What's your favorite color?

Red○
Yellow○
Blue○

Company Name ▭

△ To Checkpoints for Guideline 9.

Next slide: Example for Checkpoint 10.1

Introduction | Guidelines | Checkpoints | Examples ◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

10.1 - Until user agents allow users to turn off spawned windows, do not cause pop-ups or other windows to appear and do not change the current window without informing the user.

Priority 2

Have you ever selected a link to look at a new page, then discovered that the "back" or "go back" or "previous page" control of your browser no longer seems to work? Surprise... the link you chose may have caused the new page to load in its own new window. If both pages are full-screen you may not know that you have two browser sessions on the go. The code to open the new page in its own window might look like this:

```
<A HREF="new-page.htm" TARGET="_blank">The Wonderful World of Sponge</A>
```

where TARGET=_blank tells the browser to spawn (or open) a new window for this page.

So, if your link spawns a new window, or causes another windows to "pop up" on your display, or move the focus of the system to a new FRAME or Window, then the nice thing to do is to tell the user that something like that will happen.

Visit The Wonderful World of Sponge (Please note: this link will open the page in a new browser window.)

To Checkpoints for Guideline 10.

Next slide: Example for Checkpoint 10.2

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example
Slide 78 of 120

◀ ▶

Example for Checkpoint

10.2 - Until user agents support explicit associations between labels and form controls, for all form controls with implicitly associated labels, ensure that the label is properly positioned.

Priority 2

Authors should get in the habit of explicitly associating form controls and informative labels (see the example for Checkpoint 12.4). In the meantime, Examples 1 and 2 on this page show the layout of controls that would be relatively accessible to users of screen-readers. It should not be difficult to identify which label is associated with which control.

For all form controls with labels, ensure that:

Example 1: the label is immediately following its control on the same line (allowing more than one control/label per line).

[○ Red | ○ Blue | ○ Green]

Or...
Example 2: the label is on the line before the control (with only one label and one control per line).

Orange
○
Purple
○
Magenta
○

Example 3: shows a label and control layout that could be confusing for the screen-reader user:

[Orange | Purple | Magenta]
○ ○ ○

△ To Checkpoints for Guideline 10.

Next slide: Example for Checkpoint 10.3

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 79 of 120

10.3 - Until user agents render side-by-side text correctly provide a linear text alternative (on the current page or some other) for all tables that lay out text in parallel, word-wrapped columns.

Priority 2

Here, "user agents" also include assistive technologies.

The text in the following columns will wrap differently depending on the font size, window size, and screen resolution you have selected:

Chapter 1	Chapter 2	Chapter 3	Conclusion
It was a dark and stormy night, and all through the house the literary critics were honing their red pencils to attack the unsuspecting author who started a story with "It was a dark and stormy night...".	Suddenly a shot rang out, then a shout, then a thump, then the sound of a car door slamming on someone's thumb.	Then the inspector cried, "I know who did this dastardly deed and I will reveal the perpetrator's identity this very evening, in this very room!"	And they lived happily ever after, secure in the knowledge that their secret was safe. The End

Users of some screen-readers would hear something like this:

> Chapter 1 Chapter 2 Chapter 3 Conclusion, It was a dark and suddenly a shot rang then the inspector and they lived happily stormy night, and all out, then a shout, then cried, I know who did ever after secure in...

I realize that in this case the latter may actually better literature than the original, but you get the drift.

To Checkpoints for Guideline 10.

Next slide: Example for Checkpoint 10.3 continues

Introduction Guidelines Checkpoints Examples ◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 10.3, continued.

Example
Slide 80 of 120

◀ ▶

This is what we mean by a linear text alternative!

Chapter 1
It was a dark and stormy night, and all through the house the literary critics were honing their red pencils to attack the unsuspecting author who started a story with "It was a dark and stormy night...".

Chapter 2
Suddenly a shot rang out, then a shout, then a thump, then the sound of a car door slamming on someone's thumb.

Chapter 3
Then the inspector cried, "I know who did this dastardly deed and I will reveal the perpetrator's identity this very evening, in this very room!"

Conclusion
And they lived happily ever after, secure in the knowledge that their secret was safe.

The End

Some screen-readers and some browsers may now be able to automatically linerarize simple tables like this. The WAI Evaluation and Repair Working Group has also developed a downloadable tool for linearizing tables.

▲ To Checkpoints for Guideline 10.

Next slide: Example for Checkpoint 10.4

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

10.4 - Until user agents handle empty controls correctly, include default, place-holding characters in edit boxes and text areas.

Priority 3

Why? Some older browsers do not let a keyboard user "TAB" into an entry field. In the FORM example below, an asterisk in the Name field and some default text in the Textarea allow the user to search for the placeholding text and thus provide a means of getting the insertion cursor into the field.

Your name *

Feedback Please enter your comments here:

Submit this form.

The following code creates the above FORM example:

Your name <INPUT TYPE="TEXT" NAME="Name" SIZE="50" VALUE="* ">

Feedback <TEXTAREA NAME="TextArea1" ROWS="4" COLS="50">

Please enter your comments here: </TEXTAREA>

<INPUT TYPE="submit" VALUE="Submit this form.">

To Checkpoints for Guideline 10.

Next slide: Example for Checkpoint 10.5

Introduction Guidelines Checkpoints Examples ◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 82 of 120

◀ ▶

10.5 - Until user agents render adjacent links distinctly, include non-link, printable characters (surrounded by spaces) between adjacent links.

Priority 3

The code:

```
[ <A HREF="a.htm">Section A</A> | <A HREF="b.htm">Section B</A> | <A HREF="c.htm">Section C</A> | <A
HREF="d.htm">Section D</A> | <A HREF="e.htm">Section E</A> ]
```

produces the following horizontal list of text links:

[Section A | Section B | Section C | Section D | Section E]

The non-linked characters may cause screen-readers to pause, thus helping the user to distinguish where one link phrase ends and the next begins, and the visual aspect may help people with some visual or cognitive disabilities identify the break between links.

▲ To Checkpoints for Guideline 10.

Next slide: Example for Checkpoint 11.1

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 83 of 120

11.1 - Use W3C technologies and use the latest versions when they are supported by browsers.

Priority 2

Why W3C technologies? First and foremost, because all emerging W3C technologies will undergo a thorough review for accessibility issues. This is a commitment that immediately places W3C technologies at the forefront of many other emerging technologies.

At the time of this writing, several of the latest HTML 4.0 attributes that may significantly increase accessibility of Web pages have not been implemented in user agents. Therefore, authors may be asking:

Why should I use these attributes or elements if they aren't supported?

The authors believe that many of the new attributes and elements will be supported in the very near future, as companies release new versions of their products. Pages written to be accessible now will be ready when the technology catches up. This is the reverse of the "old" system where people with disabilities always had to play "catch up" to the new technology.

To Checkpoints for Guideline 11.

Next slide: Example for Checkpoint 11.1 continues

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint 11.1 - continued

Example
Slide 84 of 120

◀ ▶

Another question authors may be asking:

Will using these attributes or elements break my pages as they are displayed today?

User agents are supposed to ignore HTML attributes they don't support.

For example, the HTML 4.0 <A> element has an attribute, "charset" that lets you specify the ISO character set of the linked document, but no browser currently supports it. Therefore, it is simply ignored. This means that you don't have to worry about what will happen if you include an attribute that is not yet supported. The WAI hopes that user agent manufacturers will soon implement all the HTML 4.0 attributes.

▲ To Checkpoints for Guideline 11.

Next slide: Example for Checkpoint 11.1 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 11.1 - continued

Example
Slide 85 of 120

Will using these attributes or elements break my pages as they are displayed today?

User agents are also supposed to render the content (the markup between the start and end tags) of unsupported elements.

In this case, lets pretend that HTML has an element called <SHAKE> that makes an image appear to vibrate quickly. We know this is not supported by any browser. So, browsers are expected to display the content or markup the author puts between the start and end tag of the element. For example:

```
<SHAKE src="martini.gif" frequency="puree"> This martini should be shaken, not stirred. <SHAKE>
```

Since <SHAKE> is not supported, then the content is displayed:

This martini should be shaken, not stirred.

Why do we encourage you to use the new features included in W3C Recommendations (like the HTML 4.0 elements and attributes ABBR, OPTGROUP and longdesc) even if they aren't presently supported by common browsers or aren't backwards compatible with older browsers?

The previous example shows that the author can ensure that users of older technology will get the "meat" by including equivalent markup in the content of the element. It will also ensure that users of the "latest and greatest" will benefit from the use of the new element as soon as it is supported.

Please note: SHAKE is not an element of any W3C language specification and therefore this page will not validate. This example is not meant to encourage you to create your own markup.

To Checkpoints for Guideline 11.

Next slide: Example for Checkpoint 11.1 continues

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint 11.1, continued.

Example
Slide 86 of 120

Other questions that authors may be asking are:

Which browsers support these features? Which don't? There are also other issues with backward compatibility. We require that pages transform gracefully across browsers, but how can one author test all possible browser and user scenarios?

Please see User Agent Support for Accessibility. This page documents user agent support for accessibility features discussed in WCAG 1.0.

To Checkpoints for Guideline 11.

Next slide: Example for Checkpoint 11.2

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

11.2 - Avoid deprecated features of W3C technologies.

Example
Slide 87 of 120

◀ ▶

Priority 2

A deprecated element or attribute is one that has been outdated by newer constructs. Deprecated elements may become obsolete in future versions of HTML.

It is important to avoid deprecated elements so that your Web page is compliant with current browser technology. Using deprecated elements instead of newer elements will make your site incompatible with the latest version of HTML and may, despite your best intentions, decrease its accessibility.

The presentational elements (like FONT) and color attributes have been deprecated in order to encourage authors to use style sheets, thus separating page style from page content. For example, the new OBJECT element is far more versatile at handling content from different sources than the deprecated APPLET, and it is designed to degrade more gracefully if coded properly.

▲ To Checkpoints for Guideline 11.

Next slide: Example for Checkpoint 11.2 continues

Introduction Guidelines Checkpoints Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 11.2 - Continued.

Example
Slide 88 of 120

◀ ▶

Note: by convention, HTML ELEMENT names are shown in upper-case letters while the names of attributes are in lowercase letters.

Deprecated Structural Elements

- List element: DIR
  and in any list element the attributes: start and value
- Link element: APPLET
- Forms and keyboard controls: ISINDEX

Deprecated Presentational Elements

- Block alignment: CENTER
  and alignment attributes in most elements: align, valign, clear, nowrap
  Note: within tables, only align is deprecated for the overall placement of the table on a page, and of the caption if any.
- Spacing attributes: hspace, vspace, compact, type
- Text style and fonts elements: STRIKE, S, U, FONT, BASEFONT
  and the attributes: face, size
- All color attributes: background, bgcolor, color, text, link, alink, vlink
- And the noshade attribute of the HR element.

A complete list of deprecated elements is available in the HTML 4.0 specification.

▲ To Checkpoints for Guideline 11.

Next slide: Example for Checkpoint 11.3

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

11.3 - Provide information so that users may receive documents according to their preferences

Priority 3

1. Instead of including links such as "Here is the French version of this document", use content negotiation so that the French version is served to clients requesting French versions of documents.
2. In HTML 4.0 most elements allow you to specify the language of the content with the 'lang=" attribute.
3. If it is not possible to use content negotiation, in HTML use 'hreflang' with the <A> or <LINK> elements to identify the language of the target documents.

See the W3C Internationalization document on HTTP Content Negotiation for more information.

To Checkpoints for Guideline 11.

Next slide: Example for Checkpoint 11.4

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 90 of 120

11.4 - If, after best efforts, you cannot create an accessible page, provide a link to an alternative page that uses W3C technologies, is accessible, has equivalent information (or functionality), and is updated as often as the inaccessible (original) page.

Priority 1

Because of the difficulty in keeping alternative pages up to date with the full content of the original page, alternative pages should be provided only after you have tried all of the other pertinent techniques outlined in the Web Content Accessibility Guidelines to make your original page accessible (unless the alternative page is automatically generated from the same source as the original page).

Here is one way you might give visitors the choice:

Welcome to the Organization's Web Site!
Follow this link if you want the
<A HREF="path1/">dazzling but confusing site,</A>,
or follow this link if you want the
<A HREF="path2/">accessible version</A>.

When the user chooses a link, the appropriate page will be served.

Please remember that this checkpoint should only be used as a LAST RESORT. As more browsers support the latest versions of HTML, XML, CSS1 and 2 and other accessible W3C languages, pages that cannot be made accessible by following the other guidelines will become rarer.

To Checkpoints for Guideline 11.

Next slide: Example for Checkpoint 12.1

Introduction Guidelines Checkpoints Examples ◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example for Checkpoint

12.1 - Title each frame to facilitate frame identification and navigation.

Example
Slide 91 of 120

◀ ▶

Frame text

Document A

Document B

Document C

Priority 1

There are four frames in this example. The frame along the top of the page is for the page title and primary navigation bar. The narrow frame (center, left) is to be used for the Table of Contents. The largest frame (center, right) is where the content of the document chosen from the Table of Contents will appear. The frame across the bottom of the page is for the copyright, secondary navigation and other constant information. The code for the FRAMESET show the appropriate use of the title attribute.

```
<FRAMESET ROWS="20%,*,30%">
<FRAME SRC="f1.htm" title="Title and main navigation bar frame">
<FRAMESET COLS="20%,*,">
<FRAME SRC="f2.htm" title="Table of Contents frame">
<FRAME SRC="f3.htm" title="Content viewing frame">
</FRAMESET>
<FRAME SRC="f4.htm" title="Copyright, acknowlegment and secondary navigation frame">
</FRAMESET>
```

▲ To Checkpoints for Guideline 12.

Introduction Guidelines Checkpoints Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example for Checkpoint

Example
Slide 92 of 120

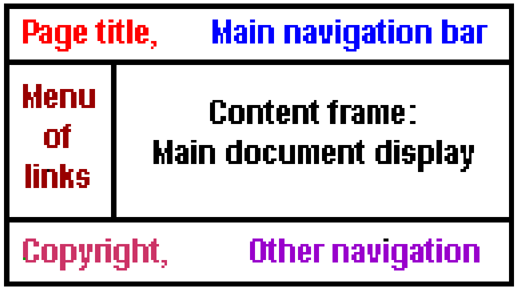
12.2 - Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone.

Priority 2

In the previous example, Checkpoint 12.1, we used the "title" attribute of the FRAME element to briefly explain the function of each frame. In some cases, you might feel that the purpose or content of your FRAME or FRAMESET may be too complicated to explain in just a few words. In those cases, we encourage you to use the "longdesc" attribute of the FRAME element to provide a link to a document that contains a complete description of the complex frame, or of the relationships between the various frames, e.g.:

<FRAME src="main.htm" longdesc="maindesc.htm" title="Main content frame.">

The "longdesc" file for this particular FRAME design might look like this:



Page title, Main navigation bar

Menu of links

Content frame: Main document display

Copyright, Other navigation

There are four frames in this example. The frame along the top of the page is for the page title and primary navigation bar. The narrow frame (center, left) is to be used for the List of links. The largest frame (center, right) is where the content of the document chosen from the List of links will appear. The frame across the bottom of the page is for the copyright, secondary navigation and other constant information.

To Checkpoints for Guideline 12.

Next slide: Example for Checkpoint 12.3

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 93 of 120

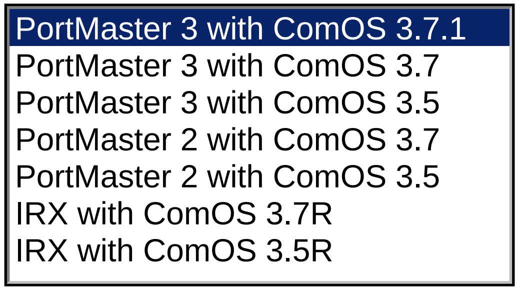
◀ ▶

12.3 - Divide large blocks of information into more manageable groups where natural and appropriate.

Priority 2

For example, in HTML, use OPTGROUP to group OPTION elements inside a SELECT element list; group form controls with FIELDSET and LEGEND; use nested lists where appropriate; use headings to structure documents, etc.

A FORM SELECT list with seven items will look like this (i.e, a simple, unstructured list) in most browsers as of March 2000.



PortMaster 3 with ComOS 3.7.1
PortMaster 3 with ComOS 3.7
PortMaster 3 with ComOS 3.5
PortMaster 2 with ComOS 3.7
PortMaster 2 with ComOS 3.5
IRX with ComOS 3.7R
IRX with ComOS 3.5R

The next page shows what the above list might look like when the OPTGROUP element is supported by browsers.

To Checkpoints for Guideline 12.

Next slide: Example for Checkpoint 12.3 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 12.3, continued.

Example
Slide 94 of 120

When browsers support the <OPTGROUP> element, the same list might look like this:

Comm Servers

Comm_Servers

Portmaster 3

PortMaster 3 with ComOS 3.7.1
PortMaster 3 with ComOS 3.7
PortMaster 3 with ComOS 3.5

Portmaster 2

PortMaster 2 with ComOS 3.7
PortMaster 2 with ComOS 3.5

Routers

IRX

IRX with ComOS 3.7R
IRX with ComOS 3.5R

Select this link to see the HTML mark-up that might create such a list in the future,

Access note: This example was created using style sheet commands to indent the text of the list to approximate how OPTGROUP will be displayed when it is supported. OPTGROUP performs some visual formatting to make it easier to distinguish between related and differing items in a list. Some screen readers may also provide auditory cues to help users distinguish between groups. OPTGROUP allows you to label your divisions with descriptive text to further help structure your information.

To Checkpoints for Guideline 12.

Next slide: Example for Checkpoint 12.3 continues

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint 12.3, continued.

Example
Slide 95 of 120

Group form controls (using the FIELDSET and LEGEND elements).

The FIELDSET element is used to group related form controls in a logical manner. A graphical browser might emphasize this relationship by drawing a border around the related elements. A future assistive device or user agent might help the user to make sense out of complex forms by enabling navigation within the group or groups.

By associating a meaningful LEGEND with each FIELDSET group, the author provides a label that will clearly explain the purpose or nature of the groupings. User agents that support LEGEND will be able to make that information available to those who want it. Some graphical browser currently show the legend as text appearing in a space in the border drawn by the FIELDSET element.

Select your options

Color
○ Red
○ Blue
○ Green
○ Gold

Engine/Powertrain Package
○ 4 Cylinder
○ 6 Cylinder
○ 6 Cylinder, All Wheel Drive

Options Required
□ Power windows
□ Heated side view mirrors
□ Moon roof
□ Anti-lock brakes

Accessibility note: If your graphical browser does not show the above example with borders and text around related controls, then it doesn't support the FIELDSET and LEGEND elements.

Select this link to see the HTML mark-up that created this example.

To Checkpoints for Guideline 12.

Next slide: Example for Checkpoint 12.4

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

12.4 - Associate labels explicitly with their controls.

Example
Slide 96 of 120

Priority 2

Following this checkpoint means that the (non-visual) user will be able to identify the correct label for a control even if the label is not located such that it is immediately obvious that it corresponds to a particular control. Future screen readers or other user agents will be able to offer this extra information about form controls to the user to aid their navigation and understanding of complex forms.

The following example lays out four checkbox controls and their text labels in such a way that the user of a screen reader might have difficulty determining which label is associated with a particular checkbox.

Options Required

□
Anti-lock brakes
Moon roof □
□ Heated side view mirrors
Power windows
□

View the code that created this example to see how the LABEL element can be use to overcome this barrier.

It will not hurt your design to start using the LABEL element now, and your page will be more accessible when agents do start supporting it. In the meantime, refer to the example for Checkpoint 10.2 for an accessible fix.

To Checkpoints for Guideline 12.

Next slide: Example for Checkpoint 13.1

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

13.1 - Clearly identify the target of each link.

Example
Slide 97 of 120

◀ ▶

Priority 2

Link text should be meaningful enough to make sense when read out of context -- either on its own or as part of a sequence of links. Link text should also be terse.

The following list shows three possible ways of choosing link text from a sentence:

1. Win a prize donated by one of our thousands of sponsors. ☺

The link phrase, "win a prize", is terse, meaningful out of context and likely to grab your attention if seen by itself.

2. Win a prize donated by one of our thousands of sponsors. ☹

Selecting the entire sentence as the link is unnecessary, and long links can wrap on the screen, which may confuse some users.

3. Click here to win a prize donated by one of our thousands of sponsors. 😠

Where is "here", why should I "click here", and what if I don't "click" but use the keyboard to select a link?.

Another thing you can do to increase accessibility is to use the title attribute on the link element to provide even more information, e.g.:

```
<A href="next.htm" title="All about income tax. ">Go to the next page.</A>
```

△ To Checkpoints for Guideline 13.

Next slide: Example for Checkpoint 13.2

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example
Slide 98 of 120

◀ ▶

Example for Checkpoint

13.2 - Provide metadata to add semantic information to pages and sites.

Priority 2

> One of the major issues of the World Wide Web as it exists today is that it is really hard to automate any tasks that one has to perform on the web. So far, the web is mainly built as a forum for human interaction; because most web documents are written for human consumption, the only available form of searching on the web (for example) is to simply match words or sentences contained in documents. Anyone who has used a web search service like AltaVista or HotBot knows that typing in a few keywords and receiving a couple of thousand "hits" is not necessarily very useful. A lot of manual "weeding" of information has to happen after that; it may also happen that the keywords for which you are searching are not prominent in the relevant document itself.
>
> A possible solution for the search problem - and for the general issue of letting automated "agents" roam the web performing useful tasks - is to provide a mechanism that allows a more precise description of things on the web. This, in turn, could elevate the status of the web from machine-readable to something we might call machine-understandable.
>
> Metadata is "data about data" or specifically in our current context "data describing web resources." The distinction between "data" and "metadata" is not an absolute one; it is a distinction created primarily by a particular application ("one application's metadata is another application's data").

From: Introduction to RDF Metadata - W3C NOTE 1997-11-13
http://www.w3.org/TR/NOTE-rdf-simple-intro

▲ To Checkpoints for Guideline 13.

Next slide: Example for Checkpoint 13.2 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 13.2, continued.

Example
Slide 99 of 120

◀ ▶

For example, use the W3C's RDF to indicate the document's author, the type of content, etc.

The Resource Description Framework (RDF) integrates a variety of web-based metadata activities including sitemaps, content ratings, stream channel definitions, search engine data collection (web crawling), digital library collections, and distributed authoring, using XML as an interchange syntax. Using RDF as a framework, you can integrate different metadata schemes (such as Dublin Core and Government Information Locator Service (GILS), and others) into one package. To learn more about RDF, please visit the W3C RDF home page for an excellent list of resources.

Another example of machine-readable metadata:
Some HTML user agents - recent versions of the LYNX text browser for instance - can build navigation tools from document relations described by the HTML LINK element and "rel" or "rev" attributes (e.g., rel="next", rel="previous", rel="index", etc.), e.g.:

```
<HEAD>
<link rel="prev" href="chk9-0.htm">
<link rel="contents" href="overchk.htm">
<link rel="next" href="chk11-0.htm">
</HEAD>
```

LYNX will build a small menu of links at the top of the page that looks like this:

# prev contents next

▲ To Checkpoints for Guideline 13.

Next slide: Example for Checkpoint 13.3

Introduction Guidelines Checkpoints Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 100 of 120

13.3 - Provide information about the general layout of a site (e.g., a site map, or table of contents).

◀ ▶

Priority 2

A textual site map (or table of contents) like the one below is a good bet because it is relatively easy to maintain (and doesn't require an accessible alternative as would a graphical map). Note that the linked text should correspond to the main page title or H1 header on the target page so that the user isn't confused by seeing different titles.

```
Welcome Page
    Home page (main page)
        Products
            Systems
                System 1000
                System 2000
                System 3000
            CPUs
                Home/Office
                Power users
            Monitors
            Peripherals
        Services
            System design
            System analysis
            Information management
              .
              .
      and so on...
```

For another example, each slide set in this curriculum also has a "Table of Contents". Links to them are included near the bottom of this page.

△ To Checkpoints for Guideline 13.

Next slide: Example for Checkpoint 13.3 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C Web Accessibility initiative

Example for Checkpoint 13.3, continued.

Example
Slide 101 of 120

It is a good idea to show some of the visible structure of the site (i.e. the page hierarchy) to help visitors orient themselves. For non-visual users, an indication of the hierarchy could be given in the TITLE attribute of the ANCHOR element, <A>:

```
<A HREF="a.htm" TITLE="level 1">Welcome Page</A>
<A HREF="b.htm" TITLE="level 2">Home page</A> (main page)
<A HREF="c1.htm" TITLE="level 3">Products</A>
<A HREF="d1.htm" TITLE="level 4">Systems</A>
<A HREF="e1.htm" TITLE="level 5">System 1000</A>
```

To Checkpoints for Guideline 13.

Next slide: Example for Checkpoint 13.4

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example
Slide 102 of 120

◀ ▶

Example for Checkpoint

13.4 - Use navigation mechanisms in a consistent manner.

Priority 2

A consistent style of presentation on each page allows users to easily find navigation buttons between pages, as well as find the primary content for each page. While this helps make it easier for everyone, it especially benefits people with learning and reading disabilities. Making it easy to predict where the needed information is found on each page will increase the likelihood that it will be found.

In case you hadn't noticed, all of the slide sets in this curriculum share a common style, with the overall layout of navigation and content elements being very similar. However, each slide set (Introduction, Guidelines, Checkpoints and Examples) includes differing visual and textual clues to help you identify and remember which set you are currently browsing.

For example, the background color of the marginal space on the Example slides is a very light purple and the main heading contains the words "'Example for...", while Checkpoint slides have a light green background and the word "Checkpoint in each main heading, and so on.

▲ To Checkpoints for Guideline 13.

Next slide: Example for Checkpoint 13.5

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example
Slide 103 of 120

◀ ▶

Example for Checkpoint

13.5 - Provide navigation bars to highlight and give access to the navigation mechanism.

Priority 3

With few exceptions, all the slides in the WCAG Curriculum provide one or more navigation bars that group the navigation features.

At the top right of each page is the bar - ◀ ▶ - that permits navigation to the previous and next page in the slide set.

Near the bottom of every page is a navigation bar -

Introduction | Guidelines | Checkpoints | Examples

that permits quick access to the Table of Contents slide of each of the major slide sets (Introduction, Guidelines, Checkpoint, and Examples).

Once you become familiar with the consistent layout and design, finding your way around these slide sets can be a breeze! This, of course, is the goal of following Checkpoint 13.5

Another benefit of knowing that important links are grouped together: you can skip over them quickly to find other important content on the page. See the next slide for details of this technique.

▲ To Checkpoints for Guideline 13.

Next slide: Example for Checkpoint 13.6

Introduction | Guidelines | Checkpoints | Examples ◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

13.6 - Group related links, identify the group (for user agents) and, until user agents do so, provide a way to bypass the group.

Priority 3

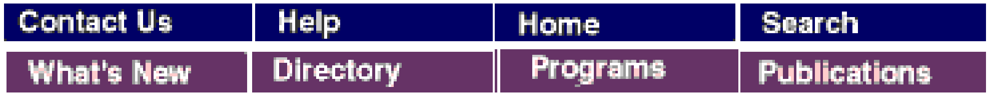
Users who are blind often jump from link to link when skimming a page or looking for information. When they do this, the "link text" (text between the <A> and </A>) is read. A navigation bar is often the first thing encountered on a page. For speech output users, this usually means having to hear a large number of similar links read on every page before they arrive at the unique content.

However when links are grouped into logical sets, such as in a navigation bar, they may be dealt with as a unit rather than as several pieces. Thus, you can precede the grouped items with a link that will skip over the set and allow the user to start reading at the beginning of the main body of the page. This is similar to how people with vision skip reading the links when they see the same set on each page.

Example 1: a text link appearing before the navigation bar:

Skip over these navigation links.

Note: these links don't go anywhere.



This line is the target of the link and would mark the start of the text following the navigation bar.

To Checkpoints for Guideline 13.

Next slide: Example for Checkpoint 13.6 continues

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint 13.6, continued

Example
Slide 105 of 120

◀ ▶

The Guidelines and User Agent Working Groups will soon recommend a "standard" way to use markup to identify groups of related links. Once a method is found, developers of user agents will be able to create browsers that automatically recognize the groups and give the user the option of skipping (or hiding) the links or presenting them normally. One such method is to package related links within the MAP element. Note that links within the MAP element do not have to be image maps (which is what most of us use MAP for).

Navigate the site.
[Section A | Section B | Section C | Section D]

The code that produces the above group of links is:

```
<MAP name="map2">
Navigate the site.<BR>
[ <A href="a.htm">Section A</A> |
<A href="b.htm">Section B</A> |
<A href="c.htm">Section C</A> |
<A href="d.htm">Section D</A> ]
</MAP>
```

▲ To Checkpoints for Guideline 13.

Next slide: Example for Checkpoint 13.6 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 13.6, continued.

Example
Slide 106 of 120

In HTML, the DIV, and FRAME elements could also be used to group links. Since these HTML 4.0 elements can take the "id" and "class" attributes, the grouping could be identified using one or more of those attributes with a descriptive name. However, unless a standard method of grouping and identifying links is agreed upon, then User Agents will have difficulty catching and rendering that information.

(See Example for checkpoint 9.5 - keyboard alternatives and Example for checkpoint 13.10 - skipping ASCII art for other examples of grouping and bypassing related elements.

To Checkpoints for Guideline 13.

Next slide: Example for Checkpoint 13.7

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 107 of 120

13.7 - Enable different types of searches for different skill levels and preferences.

Priority 3

This checkpoint does not mean that you must have some searching capability on your site. However, if you do, then here is what you need to know to make that feature as accessible as possible:

For some visitors, a simple KEYWORD search will suffice.

(**NOTE:** this is not a real search form. Do not try to submit a search.)

Enter the keyword(s)

Submit your search

And describe how the simple search routine works, e.g.
This keyword search will find all documents that contain the word you entered. If you entered more than one word, the search will find all documents that contain any of the words.

To Checkpoints for Guideline 13.

Next slide: Example for Checkpoint 13.7 continues

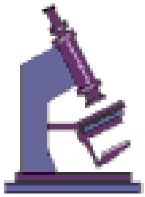
Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint 13.7, continued.

Example
Slide 108 of 120

◀ ▶



Some people need more complex search engines.

For them, provide a search tool that can handle complex Boolean searches, e.g.:

find all documents that contain WORD 1 and WORD 2 but not PHRASE 3 within 100 words of WORD 4 in documents created between DATE 1 and DATE 2

Some search engines allow the user to type the keywords and the logical operators in a script. Some have created a form-like interface where the user enters keywords and selects the operators from collections of radio buttons, check boxes or selection lists.

Both methods require that clear examples be provided to help users determine how best to do their searches. (Remember to design form-based applications that are accessible!)

Another possibility: if someone does not spell well (either because of a learning disability or the language of the search is not their primary language), a spell checker or some sort of choice of words could be provided if an entered word was not found.

△ To Checkpoints for Guideline 13.

Next slide: Example for Checkpoint 13.7 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 13.7, continued.

Example
Slide 109 of 120

Another type of search strategy is the "expanding menu" approach taken by search sites like Yahoo. In one sense, if your site's menu hierarchy is designed clearly and logically, it should function very well as a search tool. A site map is essentially your entire site menu and document hierarchy displayed at once, so it too can be a valuable search tool.

To Checkpoints for Guideline 13.

Next slide: Example for Checkpoint 13.8

Introduction | Guidelines | Checkpoints | Examples

Chuck Letourneau & Geoff Freed

Example
Slide 110 of 120

◀ ▶

Example for Checkpoint

13.8 - Place distinguishing information at the beginning of headings, paragraphs, lists, etc.

Priority 3

To decrease the amount of sifting readers perform to find important information, place distinguishing information at the beginning of headings, paragraphs, lists, etc. This is commonly referred to as "front-loading" and is especially helpful for people accessing information serially.

This list "front-loads" the links (which are probably more important than their supporting text).

- Visit the Wonderful World of Sponge, if you have a moment, and if you are so inclined.
- Sea Cucumbers Galore is another interesting invertebrate site to visit.
- Starfish and Lobsters and Crabs, Oh My! is the absolute best site.

This list is not "front-loaded":

- If you have a moment, visit the Wonderful World of Sponge, if you are so inclined.
- Another interesting invertebrate site to visit is Sea Cucumbers Galore.
- But the absolute best site is Starfish and Lobsters and Crabs, Oh My!

▲ To Checkpoints for Guideline 13.

Next slide: Example for Checkpoint 13.9

Introduction Guidelines Checkpoints Examples

◀ ▶

Chuck Letourneau & Geoff Freed

W3C® Web Accessibility initiative

Example
Slide 111 of 120

Example for Checkpoint

13.9 - Provide information about document collections (i.e., documents comprising multiple pages.).

Priority 3

- The example for Checkpoint 13.2 shows one way of providing information about a sequential document - it shows how this curriculum uses the "LINK rel=next" and "LINK rel=prev" markup to indicate the URL of the next and previous pages for extra navigation information.
- This curriculum also includes a page counter indicating which page you are on in relation to the whole set. (In the top right hand corner, before the Next slide/Previous slide links.)
- A table of contents or site map also provides useful information about a collection of related documents. See any of the "overview" slides that are linked from the bottom of this page for an example.
- Use archiving tools such as zip, tar and gzip, and stuffit to create an easily downloadable package of related files or pages.

To Checkpoints for Guideline 13.

Next slide: Example for Checkpoint 13.10

Introduction Guidelines Checkpoints Examples

Chuck Letourneau & Geoff Freed

Example for Checkpoint

13.10 - Provide a means to skip over multi-line ASCII art.

Example
Slide 112 of 120

Priority 3

As we pointed out way back in Checkpoint 1.1f, ASCII art can be very annoying to some users when they encounter it with screen readers. It is a very nice gesture to include a link that will allow those users to jump over your ASCII art masterpiece.

The following is an ASCII art picture of a bearded person wearing a hat.
(Skip over ASCII art.)

```
     MMMMMMMM
 WWWWW      WWWWW
******************
   | \O/ \O/ |
   |    |    |
   \         /
    \  ---  /
      VVVVV
       VVV
```

The following code shows you one way to do this:

```
<P>
<a href="#post-art">skip over ASCII art picture</a> of a bearded person wearing a hat.
<!-- ASCII art goes here -->
<a name="post-art">First text of following main body...</a>
```

To Checkpoints for Guideline 13.

Next slide: Example for Checkpoint 14.1

Introduction Guidelines Checkpoints Examples ◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

14.1 - Use the clearest and simplest language appropriate for a site's content.

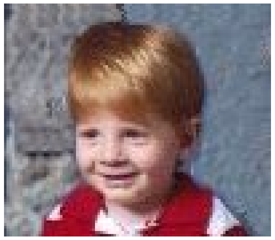
Example
Slide 113 of 120

◀ ▶

Priority 1

Refrain from obfuscating the leitmotif, Son! ☹

☺ Don't confuse the issue, Dad!



△ To Checkpoints for Guideline 14.

Next slide: Example for Checkpoint 14.1 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 14.1, continued

Example
Slide 114 of 120

◀ ▶

Follow these writing suggestions:

- Strive for clear and accurate headings and link descriptions. Review every heading, outline, and menu to see if the crucial words mean exactly what is intended, and if there are more common words that would convey the same meaning.
- State the topic of the sentence or paragraph at the beginning of the sentence or paragraph.
- Limit each paragraph to one main idea.
- Avoid idiomatic language, technical jargon, and other unfamiliar vocabulary and expressions.
- Avoid specialized meanings of familiar vocabulary, unless explanations are provided.
- Avoid the passive voice.
- Avoid complex sentence structures.
- Make link phrases terse yet meaningful enough so they make sense when read out of context, alone or as part of a series of links.

▲ To Checkpoints for Guideline 14.

Next slide: Example for Checkpoint 14.2

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

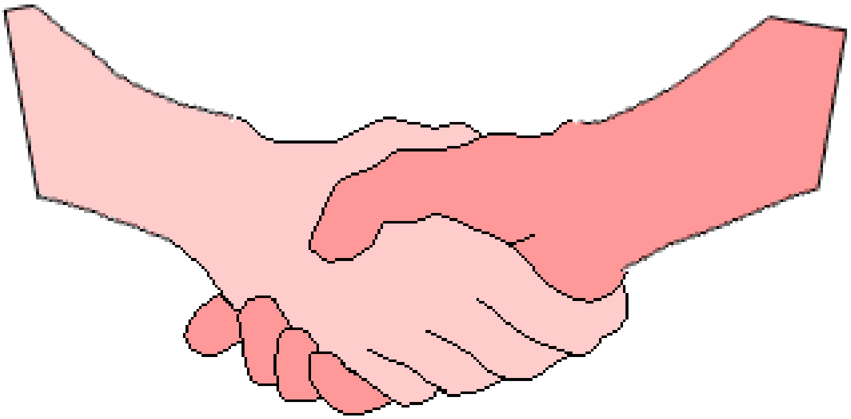
Chuck Letourneau & Geoff Freed

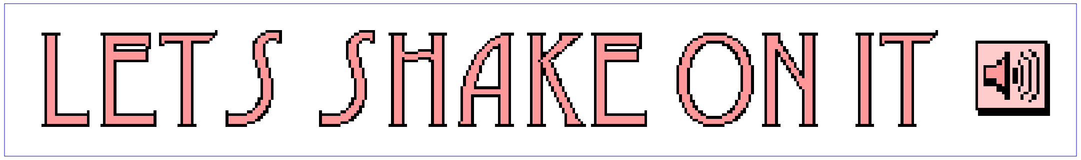
Example
Slide 115 of 120

Example for Checkpoint

14.2 - Supplement text with graphic or auditory presentations where they will facilitate comprehension of the page.

Priority 3





LETS SHAKE ON IT

In Other Words: Use pictures, sound and text to complement each other. This will help your message reach the most people.

This example uses an animated image of two people shaking hands, a text caption that complements the image, and a link to a sound clip that provides the caption in spoken form (plus the appropriate alt-text with the image files).

To Checkpoints for Guideline 14.

Next slide: Example for Checkpoint 14.2 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 14.2, continues

Example
Slide 116 of 120

Visual, non-text equivalents may include, for example images, animations, or videos. These are especially helpful for non-readers who can perceive visual presentations. For example, sighted deaf non-readers may benefit from video equivalents in manual communication (sign language). Non-readers, whether they have disability or not, may also benefit from highly graphical equivalents.

Non-visual, non-text equivalents are very diverse. Among the most common are pre-recorded audio of music, spoken language, or sound effects. Such equivalents would be especially important for non-readers who can perceive audio presentations. Presentations in the audio medium of synthesized speech and the tactile medium of braille are usually derived from text or text equivalents so usually require no additional work from the developer.

By the way, we know we shouldn't use animated images (see checkpoint 7.3) and we know that the bit-mapped text used for the caption should have been created with Cascading Style markup (see checkpoint 3.1). Would you believe "poetic license".

To Checkpoints for Guideline 14.

Next slide: Example for Checkpoint 14.3

Introduction | Guidelines | Checkpoints | Examples

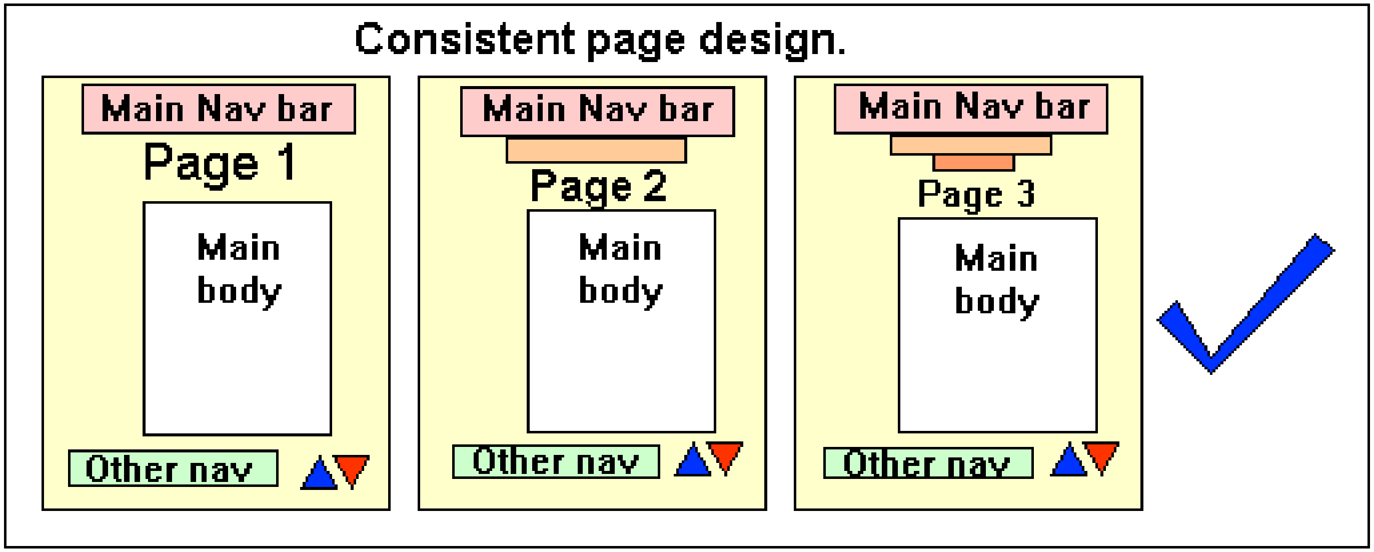
Chuck Letourneau & Geoff Freed

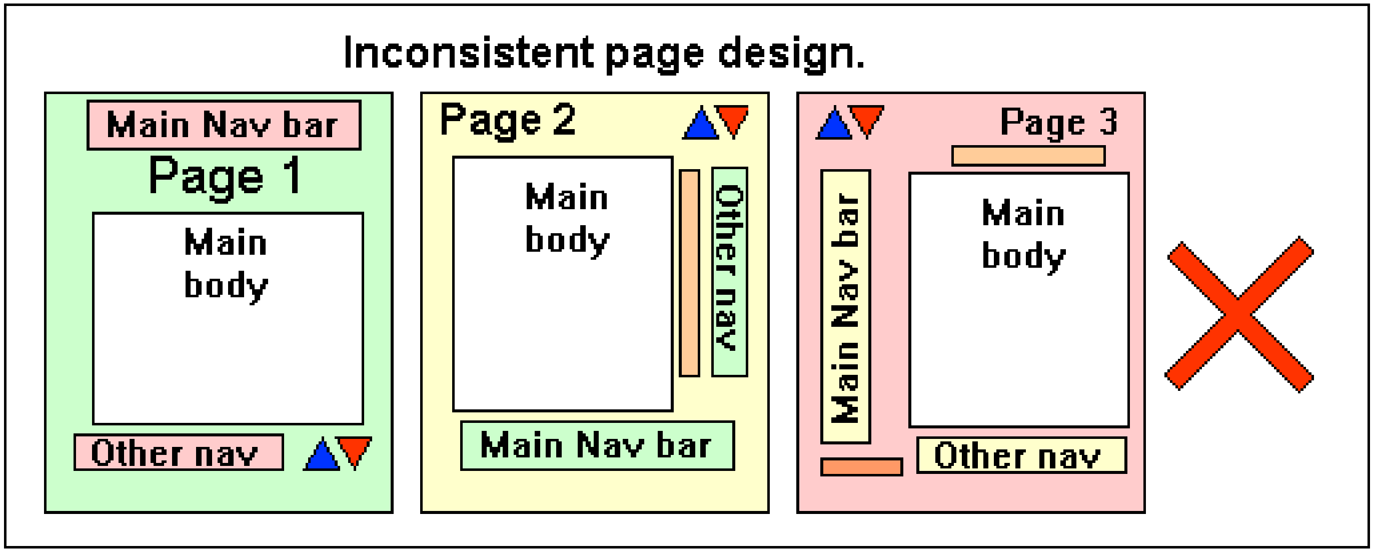
Example
Slide 117 of 120

Example for Checkpoint

14.3 - Create a style of presentation that is consistent across pages.

Priority 3





To Checkpoints for Guideline 14.

Next slide: example for Checkpoint 14.3 continues

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint 14.3, continued,

Example
Slide 118 of 120

◀ ▶

Benefits:

- The main navigation buttons are in the same location on each page. After becoming familiar with the site, you know exactly how to locate them quickly, and what actions they perform.
- The colors, text styles, heading locations and so on are consistent... you would know immediately if you left this site.
- Identifying images and supplementary text and other navigation or end-of-page features are consistently placed. When you encounter them you know you have reached the end of main content of the page.

Annoyances:

- The site might seem monotonous after browsing through many pages.
- To maintain a sense of where they are in a site, a visitor will have to rely on less obvious cues or clues on a page to keep from being confused .

Another example for this technique is this set of curriculum slides. Granted, these are designed to emulate a slide presentation, but both the beneficial (and annoying) elements of consistent design are certainly demonstrated here. We feel the benefits outweigh the annoyances.

▲ To Checkpoints for Guideline 14.

Next slide: Appendix A

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Appendix A: Validation

Example
Slide 119 of 120

◀ ▶

Automated validation methods are generally rapid and convenient but cannot identify all accessibility issues. Human review can help ensure clarity of language and ease of navigation.

Begin using validation methods at the earliest stages of development. Accessibility issues identified early are easier to correct and avoid.

Following are some important validation methods.

1. Use an automated accessibility tool and browser validation tool such as Bobby. Please note that software tools do not address all accessibility issues, such as the meaningfulness of link text, the applicability of a text equivalent, etc.
2. Validate the HTML using a service such as the W3C's HTML Validator (HTMLVAL), which checks HTML documents for compliance with W3C HTML Recommendations and other HTML standards.
3. Validate style sheets using a service such as the W3C CSS Validation Service.
4. Use a text-only browser, such as Lynx or an emulator, such as Lynx-Me or Lynx Viewer.
5. Use multiple graphic browsers with:
   - sounds and graphics loaded,
   - graphics not loaded,
   - sounds not loaded,
   - no mouse,
   - frames, scripts, style sheets, and applets not loaded
6. Use several browsers, old and new.
7. Use a self-voicing browser, such as pwWebSpeak or Home Page Reader, a screen reader, such as JAWS or outSPOKEN, magnification software, a small display, etc.
8. Use spell and grammar checkers. A person reading a page with a speech synthesizer may not be able to decipher the synthesizer's best guess for a word with a spelling error. Eliminating grammar problems increases comprehension.
9. Review content and structure for clarity and simplicity.
10. Invite people with disabilities to review your documents. Expert and novice users with disabilities will provide valuable

feedback about accessibility or usability problems and their severity.

△ To Checkpoints for Appendix A.

Next slide: End of Example set.

Introduction | Guidelines | Checkpoints | Examples

◀ ▶

Chuck Letourneau & Geoff Freed

Example for Checkpoint

Example
Slide 120 of 120

◀

End of the Example set.

Where to next?

- go to the Introductory Set
- go to the Guideline Set
- go to the Checkpoint Set
- go back to the start of the Example set
- or select from the Table of Contents links below

Any comments or questions?

- about this curriculum? Contact the authors (follow the links at the bottom of this page
- about the guidelines or techniques? See the WAI Home page for more information.

Introduction | Guidelines | Checkpoints | Examples

◀

Chuck Letourneau & Geoff Freed

Introduction: WAI Web Content Accessibility Curriculum

by Chuck Letourneau & Geoff Freed

Table of contents

About the Authors

Geoff Freed

Geoff is the project manager of the Web Access Project at the CPB/WGBH National Center for Accessible Media (NCAM) at the WGBH Educational Foundation in Boston, MA. Geoff has a long history in access technology, beginning as a caption writer at The Caption Center in 1985 and, in 1989, as the manager of external captioning projects. Upon joining NCAM in 1991, Geoff supervised the creation and administration of CC University, a program to teach public television stations to caption their own local programming. Geoff began work on the Web Access Project in 1996, developing and testing methods to make the Web and multimedia more accessible to users who are blind or deaf.

Geoff can be reached via e-mail at Geoff_Freed@wgbh.org

Use the BACK, GO BACK or PREVIOUS PAGE feature of your browser to return.

Chuck Letourneau

Chuck is the president of and principal consultant for Starling Access Services, a company that specializes in accessible Web site design, site evaluations, and training. Chuck has been involved with the Web Accessibility Initiative since its official launch in April, 1997. He is co-chair (with Dr. Gregg Vanderheiden of the Trace R&D Center) of the WAI Page Author Guidelines working group, and is very active in the Education and Outreach working group. Chuck is also the secretary of the Assistive Devices Industry Association of Canada.

Chuck can be reached via e-mail at cpl@starlingweb.com.

More information about Starling Access Services can be found on their Web site.

Use the BACK, GO BACK or PREVIOUS PAGE feature of your browser to return.

Author's slide 1 of 1

Guidelines: WAI Web Content Accessibility Curriculum

by Chuck Letourneau & Geoff Freed

Table of contents

W3C® Web Accessibility initiative

Checkpoints: WAI Web Content Accessibility Curriculum

by *Chuck Letourneau* & *Geoff Freed*

Table of contents

Examples: WAI Web Content Accessibility Curriculum

by Chuck Letourneau & Geoff Freed

Table of contents

- 1.3 - Until user agents can automatically read aloud the text equivalent of a visual track, provide an auditory description of the important information of the visual track of a multimedia presentation.
- 1.3 - continued
- 1.4 - For any time-based multimedia presentation, synchronize equivalent alternatives with the presentation.
- 1.4 - continued
- 1.5 - Until user agents render text equivalents for client-side image map links, provide redundant text links for each active region of a client-side image map.
- 2.1 - Ensure that all information conveyed with color is also available without color, for example from context or markup.
- 2.2 - Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen.
- 2.2, continued.
- 3.1 - When an appropriate markup language exists, use markup rather than images to convey information.
- 3.2 - Create documents that validate to published formal grammars.
- 3.3 - Use style sheets to control layout and presentation.
- 3.3, continued.
- 3.4 - Use relative rather than absolute units in markup language attribute values and style sheet property values.
- 3.5 - Use header elements to convey logical structure and use them according to specification.
- 3.5, continued
- 3.6 - Mark up lists and list items properly.
- 3.6, continued.
- 3.7 - Mark up quotations. Do not use quotation markup for formatting effects such as indentation.
- 3.7, continued.
- 3.7, continued.
- 4.1 - Clearly identify changes in the natural language of a document's text and any text equivalents
- 4.2 - Specify the expansion of each abbreviation or acronym in a document where it first occurs.
- 4.3 - Identify the primary natural language of a document.

- 5.1 - For data tables, identify row and column headers.
- 5.1, continued.
- 5.2 - For data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells.
- 5.2, continued.
- 5.2, continued.
- 5.3 - Do not use tables for layout unless the table makes sense when linearized.
- 5.4 - If a table is used for layout, do not use any structural markup for the purpose of visual formatting.
- 5.5 - Provide summaries for tables.
- 5.6 - Provide abbreviations for header labels.
- 6.1 - Organize documents so they may be read without style sheets. When an HTML document is rendered without associated style sheets, it must still be possible to read the document.
- 6.1, continued.
- 6.2 - Ensure that equivalents for dynamic content are updated when the dynamic content changes.
- 6.2, continued.
- 6.3 - Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page.
- 6.4 - For scripts and applets, ensure that event handlers are input device-independent.
- 6.5 - Ensure that dynamic content is accessible or provide an alternative presentation or page.
- 6.5 - continued.
- 7.1 - Until user agents allow users to control it, avoid causing the screen to flicker.
- 7.2 - Until user agents allow users to control it, avoid causing content to blink.
- 7.3 - Until user agents allow users to freeze moving content, avoid movement in pages.
- 7.4 - Until user agents provide the ability to stop the refresh do not create periodically auto-refreshing pages.
- 7.5 - Until user agents provide the ability to stop auto-redirect do not use markup to redirect pages automatically. Instead, configure the server to perform redirects.
- 8.1 - Make programmatic elements such as scripts and applets directly accessible or compatible with assistive

technologies.

- 9.1 - Provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape.
- 9.1, continued.
- 9.2 - Ensure that any element that has its own interface can be operated in a device-independent manner.
- 9.2, continued.
- 9.3 - For scripts, specify logical event handlers rather than device-dependent event handlers.
- 9.3, continued.
- 9.3, continued.
- 9.4 - Create a logical tab order through links, form controls, and objects.
- 9.4 - continued.
- 9.4, continued.
- 9.5 - Provide keyboard shortcuts to important links (including those in client-side image maps), form controls, and groups of form controls.
- 10.1 - Until user agents allow users to turn off spawned windows, do not cause pop-ups or other windows to appear and do not change the current window without informing the user.
- 10.2 - Until user agents support explicit associations between labels and form controls, for all form controls with implicitly associated labels, ensure that the label is properly positioned.
- 10.3 - Until user agents render side-by-side text correctly provide a linear text alternative (on the current page or some other) for all tables that lay out text in parallel, word-wrapped columns.
- 10.3, continued.
- 10.4 - Until user agents handle empty controls correctly, include default, place-holding characters in edit boxes and text areas.
- 10.5 - Until user agents render adjacent links distinctly, include non-link, printable characters (surrounded by spaces) between adjacent links.
- 11.1 - Use W3C technologies and use the latest versions when they are supported by browsers.
- 11.1 - continued

- 11.1 - continued
- 11.1, continued.
- 11.2 - Avoid deprecated features of W3C technologies.
- 11.2 - Continued.
- 11.3 - Provide information so that users may receive documents according to their preferences
- 11.4 - If, after best efforts, you cannot create an accessible page, provide a link to an alternative page that uses W3C technologies, is accessible, has equivalent information (or functionality), and is updated as often as the inaccessible (original) page.
- 12.1 - Title each frame to facilitate frame identification and navigation.
- 12.2 - Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone.
- 12.3 - Divide large blocks of information into more manageable groups where natural and appropriate.
- 12.3, continued.
- 12.3, continued.
- 12.4 - Associate labels explicitly with their controls.
- 13.1 - Clearly identify the target of each link.
- 13.2 - Provide metadata to add semantic information to pages and sites.
- 13.2, continued.
- 13.3 - Provide information about the general layout of a site (e.g., a site map, or table of contents).
- 13.3, continued.
- 13.4 - Use navigation mechanisms in a consistent manner.
- 13.5 - Provide navigation bars to highlight and give access to the navigation mechanism.
- 13.6 - Group related links, identify the group (for user agents) and, until user agents do so, provide a way to bypass the group.
- 13.6, continued
- 13.6, continued.
- 13.7 - Enable different types of searches for different skill levels and preferences.

Priorities of the Checkpoints

In the Web Content Accessibility Guidelines and in this curriculum, each checkpoint associated with a guideline is assigned a priority rating. The priority gives you a quick indication of the importance of following the particular checkpoint in your own page designs.

Priority 1

- An author must follow this guideline, or one or more groups of users will find it impossible to access information in the document.
- Implementing this guideline is a basic requirement for some groups to be able to use Web documents.

Please use the Back, or Go Back feature of your browser to return to the previous page.

Priority 2

- This guideline should be followed by an author, or one or more groups of users might find it difficult to access information in the document.
- Implementing this guideline will significantly improve access to Web documents.

Please use the Back, or Go Back feature of your browser to return to the previous page.

Priority 3

- This guideline may be followed by an author to make it easier for one or more groups of users to access information in the document.
- Implementing this guideline will improve access to Web documents.

Please use the Back, or Go Back feature of your browser to return to the previous page.

Example destination page.

... and you win a brief look at this dead-end page.

Please use the BACK or GO BACK command of your browser to return to the previous page.

This page is not meant to include any content. It is only here to prevent confusing broken links from many of the examples in the Curriculum.

Chuck Letourneau & Geoff Freed

Script test slide A

Section A

This page (as well as others linked from the example image maps) are not meant to include any content. They are only here to prevent confusing broken links. **Please use the BACK or GO BACK command of your browser to return to the previous page**.

Section B

This page (as well as others linked from the example image maps) are not meant to include any content. They are only here to prevent confusing broken links. **Please use the BACK or GO BACK command of your browser to return to the previous page**.

Section C

This page (as well as others linked from the example image maps) are not meant to include any content. They are only here to prevent confusing broken links. **Please use the BACK or GO BACK command of your browser to return to the previous page**.

Section D

This page (as well as others linked from the example image maps) are not meant to include any content. They are only here to prevent confusing broken links. **Please use the BACK or GO BACK command of your browser to return to the previous page**.

Empty Page

This page is not meant to include any content. It only exists to prevent confusing broken links (those "404" messages you get from a Web server).

Please use the BACK or GO BACK command of your browser to return to the previous page.

Mark-up example for Checkpoint 12.4: use of LABEL with Form controls

This code creates the example form, and highlights the use of the LABEL element:

```
<FIELDSET>

<LEGEND align="top">
Options Required
</LEGEND><BR>

<INPUT TYPE="CHECKBOX"
NAME="opt1" ID="opt1" VALUE="abs"><BR>
<LABEL for="opt1">>Anti-lock breaks</LABEL>

<LABEL for="opt2">Moon roof </LABEL>
<INPUT TYPE="CHECKBOX"
NAME="opt2" ID="opt2" VALUE="mr"><BR>
<INPUT TYPE="CHECKBOX"
NAME="opt3" ID="opt3" VALUE="hm">
<LABEL for="opt3">Heated mirrors</LABEL>

<LABEL for="opt4">Power windows</LABEL><BR>
```

```
<INPUT TYPE="CHECKBOX"

NAME="opt4" ID="opt4" VALUE="pw">


</FIELDSET>
```

Example for Checkpoint 12.4

Mark-up example for Checkpoint 12.3: use of FIELDSET and LEGEND

And the markup to group (FIELDSET) and explain (LEGEND) the groups is:

```
<FIELDSET> <LEGEND> Select your options </LEGEND>
<FIELDSET> <LEGEND>Engine/Powertrain packages </LEGEND><BR>
<INPUT TYPE="RADIO" NAME="power" VALUE="4cyl" TABINDEX="4"> 4 cylinder<BR>
<INPUT TYPE="RADIO" NAME="power" VALUE="6cyl"> 6 cylinder<BR>
<INPUT TYPE="RADIO" NAME="power" VALUE="6awd"> 6 cylinder, all wheel drive
</FIELDSET>
</FIELDSET>
```

Please note that this code example only creates a part of the example on the previous page. However, all the important features are shown here.

◀ Example for Checkpoint 12.3

Mark-up example for Checkpoint 12.3: use of OPTGROUP

And here is some code that might generate the neatly ordered selection box list simulated on the previous page. However, please note that HTML4.0 does NOTallow the nesting of OPTGROUP elements as shown in this example. The HTML 4.0 specification states:

> "Implementors are advised that future versions of HTML may extend the grouping mechanism to allow for nested groups (i.e., OPTGROUP elements may nest) as shown in the above example. This will allow authors to represent a richer hierarchy of choices."

```
<SELECT name="ComOS">
<OPTGROUP label="Comm Servers">

<OPTGROUP label="PortMaster 3">
<OPTION label="3.7.1" value="pm3_3.7.1"> PortMaster 3 with ComOS 3.7.1
<OPTION label="3.7" value="pm3_3.7"> PortMaster 3 with ComOS 3.7
<OPTION label="3.5" value="pm3_3.5"> PortMaster 3 with ComOS 3.5
</OPTGROUP>
<OPTGROUP label="PortMaster 2">
<OPTION label="3.7" value="pm2_3.7">PortMaster 2 with ComOS 3.7
<OPTION label="3.5" value="pm2_3.5">PortMaster 2 with ComOS 3.5
</OPTGROUP>

</OPTGROUP>



<OPTGROUP label="Routers">

<OPTGROUP label="IRX">
<OPTION label="3.7R" value="IRX_3.7R">IRX with ComOS 3.7R
<OPTION label="3.5R" value="IRX_3.5R"> IRX with ComOS 3.5R
</OPTGROUP>

</OPTGROUP>
```

◀ Example for Checkpoint 12.3

Priority 1

There are four frames in this example. The frame along the top of the page is for the page title and primary navigation bar. The narrow frame (center, left) is to be used for the Table of Contents. The largest frame (center, right) is where the content of the document chosen from the Table of Contents will appear. The frame across the bottom of the page is for the copyright, secondary navigation and other constant information. The code for the FRAMESET show the appropriate use of the title attribute.

```
<FRAMESET ROWS="20%,*,30%">
<FRAME SRC="f1.htm" title="Title and main navigation bar frame">
<FRAMESET COLS="20%,*,">
<FRAME SRC="f2.htm" title="Table of Contents frame">
<FRAME SRC="f3.htm" title="Content viewing frame">
</FRAMESET>
<FRAME SRC="f4.htm" title="Copyright, acknowlegment and secondary navigation frame">
</FRAMESET>
```

▲ To Checkpoints for Guideline 12.

Section E

This page (as well as others linked from the example image maps) are not meant to include any content. They are only here to prevent confusing broken links. **Please use the BACK or GO BACK command of your browser to return to the previous page**.

No Surprise...

You knew this page would be launched in a new window, which may or may not be the same size as the window you were previously viewing. If you select FILE EXIT or FILE CLOSE from the file menu of the browser, this window will shut and the page you came from is still running in its own browser window.

If you prefer...

If you prefer not to use our on-line form to respond, please use one of the following methods:

1. Send an e-mail message to us@here.org with the following information: your company name and your favourite color (red, green or blue).
2. Send us a FAX (at 1-888-555-5555) with the requested information.
3. Call us at 1-888-555-5556 and speak to our friendly customer service representative.

Please use the BACK or GO BACK command of your browser to return to the previous page.

Chuck Letourneau & Geoff Freed

Form alternative slide

Mark-up example for Checkpoint 9.4: use of TABINDEX

And the markup to produce a logical tab order through this badly organized form is:

```
<FORM METHOD="POST">

<INPUT TYPE="SUBMIT" VALUE="Enter" tabindex="6">

What's your favourite colour?<BR>
Red <INPUT TYPE="CHECKBOX" NAME="CBVar1" VALUE="CB4" tabindex="3"><BR>
Green <INPUT TYPE="CHECKBOX" NAME="CBVar2" VALUE="CB5" tabindex="4"><BR>
Blue <INPUT TYPE="CHECKBOX" NAME="CBVar3" VALUE="CkB6" tabindex="5"><BR>

Company Name <INPUT TYPE="TEXT" NAME="Com" tabindex="2"><BR>

<INPUT TYPE="RESET" VALUE="Undo" tabindex="7">

<A HREF="form-alt.htm" tabindex="1">Skip the form to find accessible alternatives</A>.<BR>

</FORM>
```

◀ Example for Checkpoint 9.4

Mark-up example for Checkpoint 6.5: use of NOFRAMES

The following code would create a two-frame layout in browsers that support frames, while the content of the NOFRAMES element would be create a single window in browsers that do not support frames.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
<HTML>
<HEAD>
<TITLE>This is top.html</TITLE>
</HEAD>
<FRAMESET cols="50%, 50%" title="Our big document">
<FRAME src="main.html" title="Where the content is displayed">
<FRAME src="table_of_contents.html" title="Table of Contents">
<NOFRAMES>
<A href="table_of_contents.html">Table of Contents.</A>
<!-- other navigational links that are available in main.html are available here also. -->
</NOFRAMES>
</FRAMESET>
</HTML>
```

◀ Example for Checkpoint 6.5

Mark-up example for Checkpoint 6.4: use of NOSCRIPT

Here is the code used to create the scripted button:

```
<BUTTON ONMOUSEOVER="pop()">
Win a prize!</BUTTON>.
```

And the code of the script function itself:

```
<SCRIPT language="JavaScript" type="text/javascript">
function pop() { alert ("You Win First Prize!") }
</SCRIPT>
<NOSCRIPT>You Win First Prize!</NOSCRIPT>
```

◀ Example for Checkpoint 6.4

Mark-up example for Checkpoint 5.2: an accessible TABLE

This code shows one way this table might be marked up to allow the greatest access:

```
<TABLE BORDER="1" CELLPADDING=2 CELLSPACING=3>
<CAPTION>TRAVEL EXPENSES (actual cost, US$)</CAPTION>

<THEAD>
<TR>
<TH><P><SPAN ID="t1-r1-l1">TRIP</SPAN>,<BR>
<SPAN ID="t1-r1-l2"> date</SPAN></P></TH>
<TH SCOPE="column">Meals</TH>
<TH SCOPE="column">Room</TH>
<TH SCOPE="column"><ABBR="Transportation">Trans.</ABBR></TH>
<TH SCOPE="column">Total</TH>
</TR>
</THEAD>

<TBODY>
<TR>
<TH SCOPE="rowgroup" HEADERS="t1-r1-l1">San Jose</TH>
</TR>

<TR>
<TD SCOPE="row" HEADERS="t1-r1-l2"> 25 Aug 97</TD>
<TD>37.74</TD>
<TD>112.00</TD>
<TD>45.00</TD>
</TR>

<TR>
<TD SCOPE="row" HEADERS="t1-r1-l2"> 26 Aug 97</TD>
```

```
<TD>27.28</TD>
<TD>112.00</TD>
<TD>45.00</TD>
</TR>

<TR>
<TD SCOPE="row">Subtotal</TD>
<TD>65.02</TD>
<TD>224.00</TD>
<TD>90.00</TD>
<TD>379.02</TD>
</TR>
</TBODY>

<TBODY>
<TR>
<TH SCOPE="rowgroup" HEADERS="t1-r1-l1">Seattle</TH>
</TR>

<TR>
<TD SCOPE="row" HEADERS="t1-r1-l2"> 27 Aug 97</TD>
<TD>96.25</TD>
<TD>109.00</TD>
<TD>36.00</TD>
</TR>

<TR>
<TD SCOPE="row" HEADERS="t1-r1-l2"> 28 Aug 97</TD>
<TD>35.00</TD>
<TD>109.00</TD>
<TD>36.00</TD>
</TR>

<TR>
<TD SCOPE="row">Subtotal</TD>
<TD>131.25</TD>
```

```
<TD>218.00</TD>
<TD>72.00</TD>
<TD>421.25</TD>
</TR>
</TBODY>

<TBODY>
<TR>
<TH SCOPE="row">Totals</TH>
<TD>196.27</TD>
<TD>442.00</TD>
<TD>162.00</TD>
<TD>800.27</TD>
</TR>
</TBODY>
</TABLE>
```

Example for Checkpoint 5.2

Mark-up example for Checkpoint 5.1: use of 'headers' attribute in a TABLE

```
<TR>
<TH id="t1">Name</TH>
<TH id="t2">Cups</TH>
<TH id="t3" abbr="Type">Type of Coffee</TH>
<TH id="t4">Sugar?</TH>
</TR>
<TR>
<TD headers="t1">T. Sexton</TD>
<TD headers="t2">10</TD>
<TD headers="t3">Espresso</TD>
<TD headers="t4">No</TD>
</TR>
<TR>
<TD headers="t1">J. Dinnen</TD>
<TD headers="t2">5</TD>
<TD headers="t3">Decaf</TD>
<TD headers="t4">Yes</TD>
</TR>
```

◀ Example for Checkpoint 5.1

Mark-up example for Checkpoint 5.1: use of 'TH' element in a TABLE

```
<TABLE border=1>
<CAPTION>Example of a simple data table
created using HTML markup.</CAPTION>
<TR>
<TD></TD>
<TH>Col. 1 header</TH>
<TH>Col. 2 header</TH>
</TR>
<TR>
<TH>Row 1 header</TH>
<TD>C1R1</TD>
<TD>C1R2</TD>
</TR>
<TR>
<TH>Row 2 header</TH>
<TD>C2R1</TD>
<TD>C2R2</TD>
</TR>
```

</TABLE>

◀ Example for Checkpoint 5.1

Mark-up example for Checkpoint 3.1: MathML

$$\int_0^t \frac{dx}{x}$$

The MathML markup for this equation might be:

```
<semantics>
<mrow>
<msubsup>
<mo>&int;</mo>
<mn>0</mn>
<mi>t</mi>
</msubsup>
<mfrac>
<mrow>
<mo>d</mo>
<mi>x</mi>
</mrow>
<mi>x</mi>
</mfrac>
```

```
</mrow>


<annotation-xml encoding="MathML-Content">

<apply>

<int/>

<bvar><ci>x</ci></bvar>

<lowlimit><cn>0</cn></lowlimit>

<uplimit><ci>t</ci></uplimit>

<apply>

<divide/>

<cn>1</cn>

<ci>x</ci>

</apply>

</apply>

</annotation-xml>
</semantics>
```

Don't panic -- there will be programs that will generate this markup for you. For more about information (and the source of this example), see the W3C MathML Recommendation.

Example for Checkpoint 3.1

Mark-up example for Checkpoint 1.5: alternative for a Client side image map

---

Some code that would create the image map and its text alternative.

```
<IMG SRC="img/imgmap1.gif"
ALT="Image map: please use the
alternative links provided with the map."
TITLE="Various icons relating to disability"
USEMAP="#map1" BORDER=0>

<MAP NAME="map1">
<AREA COORDS="0,0,39,39" HREF="a.htm"
ALT="Link to section A of the site. ">
<AREA COORDS="40,0,79,39" HREF="b.htm"
ALT="Link to section B of the site. ">
<AREA COORDS="80,0,120,39" HREF="c.htm"
ALT="Link to section C of the site. ">
<AREA COORDS="121,0,160,39" HREF="d.htm"
ALT="Link to section D of the site. ">
</MAP>

<MAP NAME="map2">
```

```
[ <A HREF="a.htm">Section A</A>

| <A HREF="b.htm">Section B</A>

| <A HREF="c.htm">Section C</A>

| <A HREF="d.htm">Section D</A> ]

</MAP>
```

Example for Checkpoint 1.5

Multimedia Software Page

Example for Checkpoint 1.3 Example for Checkpoint 1.4

In order to view the accessible multimedia presentations, you must first download the appropriate software. Your choices are listed below.

1. **QuickTime**
   MoviePlayer, which is part of QuickTime 4.0, is necessary to show accessible QuickTime movie clips using either a Macintosh or PC. You can download the free version for either platform or, if you want to take advantage of all of QuickTime's features, pay the extra $29.95 and get the upgrade to QuickTime 4.0 Pro. However, for the purposes of this demonstration, all that's needed is the free version. Once you have it installed read more information about playing the accessible QuickTime clip.

2. **SMIL**
   Different SMIL players are available. To view the examples contained in this tutorial, you'll need the G2 Player from RealNetworks. The free version of the software will suffice for our movies. You'll also need WinZip to unzip the SMIL clip prior to playing it. Download and install the G2 on your PC (no Mac version is available yet). Once you have it installed, read more information about playing the accessible SMIL clip.

3. **SAMI**
   Microsoft's SAMI format is now available for use, but authoring tools are still being developed. Watch for the release of NCAM's multiformat Media Access Generator (MAGpie) during the fall of 1999, as well as a basic SAMI editor from Microsoft. For the time being, though, download and install Media Player on your PC (no Mac version is available yet). Once you have it installed, you'll be able to play the demo SAMI clips found in these slides (coming soon!). You can read more about SAMI at Microsoft's SAMI information page.

4. **MAGpie**
   Although not necessary for viewing movie or audio clips, new software is coming which will make it easier to create accessible multimedia. NCAM's new caption-editing tool, MAGpie, will allow you to output captions in three different formats-- QuickTime, SMIL and SAMI-- from a single caption text file. MAGpie will be made available during the fall of 1999, and will be free for the downloading. Watch this space for updated information or check NCAM's Web Access Project

for details. For the time being-- until MAGpie is released-- you can visit the Web Access Project Web site for instructions on writing your own SMIL and QuickTime captions.

◀ Example for Checkpoint 1.3 ◀ Example for Checkpoint 1.4

Accessible QuickTime clips

Always include captions and audio descriptions in your QuickTime multimedia.

- QuickTime movies are playable on either a Mac or PC.
- QuickTime 4.0 and 3.0 allow the user to add captions and descriptions using either the Mac or PC. Previous versions of QuickTime allow the user to add captions and descriptions using the Mac only.
- Once the clip has been downloaded from the Web, the accessibility tracks can be toggled using the Enable Tracks feature.
- The caption track is searchable, which makes it useful for the classroom or as a database.
- More QuickTime movies that are accessible may be found at NCAM's Web site.

For an example of a properly captioned and audio-described movie, download this zipped QuickTime clip called, 'Catch!' (The file is a little over 2.4MB in size). After unzipping, use QuickTime 4.0 to play it back on either a Mac or PC.

Don't have QuickTime 4.0? Check the multimedia software page before going any further. By the way, here is the captioned and described transcript for the movie, "Catch!".

To play the movie, open MoviePlayer and then open WAICatch.mov. The movie has been uploaded with both captions and descriptions turned **on**. You can toggle the caption and/or description tracks on and off as necessary by following the simple directions below:

1. Mac:
   Open Edit menu
   Choose Enable Tracks
   Toggle the tracks on or off as appropriate
   Choose the OK button

2. PC:
   Open Edit menu
   Choose Enable Tracks
   Toggle the tracks on or off as appropriate
   Choose the OK button

MoviePlayer also lets you search the text track for specific words or phrases. To do so, open the Edit menu, choose Find and type in a word that appears in the captions. Press the Enter or Return key, and MoviePlayer will move to the first frame of the clip where the text appears. Note that this feature will work even if the text track has been turned off.

To the Multimedia Software Page.

Caption and Description Transcript for Quicktime Movie: "Catch!"

Description:
A title: "On-air Science and Math Fair. Math is Everywhere."

Caption:
Man: Go long, way out.

Description:
A man throws a football.

Caption:
Professor: Submitted for your consideration: the ball has been thrown, and you want to catch it. But how do you know where it will go so you can be there when it comes down?

One way is to use this formula.

Description:
The professor scribbles numbers on a blackboard.

Caption:
It factors in velocity, acceleration and time to calculate the distance the ball will go. But then, you knew that. Your brain estimates all of these values in the first seconds of the ball's flight to calculate where you have to go to catch the ball.

Description:
A beach ball bounces off his head. He tries to catch it.

Caption:
This math equation helps us understand the physical world and how we function in it. Whether you do it on the board or on the field, they both work. So, next time you go out for that long pass, remember: Math is everywhere.

Math is everywhere!

Description:
A title: "Math is Everywhere."

Please use the BACK, GO BACK or PREVIOUS PAGE function of your browser to return to the page you came from.

Accessible SMIL clips

Always include captions and audio descriptions in your SMIL multimedia.

- SMIL is written in XML.
- SMIL synchronizes separate media tracks (like video, captions and audio descriptions) on the fly, as opposed to embedding each of these tracks into a single large movie. This makes it easier to create and maintain multimedia.
- These movie clips contains both captions and descriptions. CAR.SMI contains captions and descriptions in both English and German, but ELEV1.SMI contains captions and descriptions only in English. The RealNetworks G2 Player has the capability to toggle the captions on and off, as described later in this slide. **Note:** The audio descriptions are open -- that is, they can't be turned off.
- These particular SMIL clips are playable only on the RealNetworks G2, which uses some proprietary code. Other SMIL players and editors are GRiNS, available from Oratrix, and SOJA, available from Helio.

For an example of how it's done, play the two movie clips shown below. Don't have a SMIL player? Check the multimedia software page before going any further.

If you want to demonstrate these examples in a classroom setting, we suggest you download the zipped files as indicated. Then the clips and associated files can be played very efficiently from your hard-drive. Also, you can look at the various files that combine to make the SMIL clip

1. Download CAR.ZIP (2 MB) and/or ELEV1.ZIP (5.5 MB).
2. Unzip each file with WinZip. (Download and install WinZip if necessary.)
3. Start the G2 player and play the movies by opening the appropriate SMIL files: CAR.SMI or ELEV1.SMI. In each movie, you should see captions centered below the video window and hear audio descriptions interspersed with the program narration.

To see what a SMIL command file looks like, view the source for either the Car clip or the Elevator clip

To see what a synchronized SMIL caption file looks like, view either CARCAPSEN.RT or ELEVCAPS.RT

Special note:

CAR.SMI has captions and descriptions in both English and German. Using the G2 player, you can toggle between these two languages. To do so...

Open the G2 Player. Before playing the movie, you'll need to program the G2 to display captions (or subtitles) and descriptions in one of two languages:

. Open the Options menu

b. Choose Preferences

c. Choose the Content tab

d. Choose English or German from the Language listbox

e. Check the Enable Captions box at the bottom of the window

1. Play the movie by opening CAR.SMI. You should see captions or subtitles centered below the video window and hear audio descriptions interspersed with the program narration. To change languages again, just repeat step C, above.
2. All movie clips need to have an accompanying transcript, so take a look at the transcript of ELEVATOR.

To the Multimedia Software Page.

```
<smil>
 <head>
                <meta name="title" content="Car"/>
                <meta name="author" content="NCAM/NOVA"/>
                <meta name="copyright" content="©1998 WGBH Educational Foundation" />
   <layout>




   <root-layout background-color="black" height="180" width="250"/>
   <region id="videoregion" background-color="black" top="5" left="35" height="130" width="180"/>
   <region id="textregion" top="135" left="-37" height="120" width="250"/>
   </layout>
 </head>

    <body>
     <par>

<!-- VIDEO -->

      <video src="carsilent.rm" region="videoregion"/>

<!-- PROGRAM AUDIO -->


     <switch>
       <audio src="carnarren.rm" system-language="en"/>
       <audio src="carnarde.rm" system-language="de"/>
 <!--  <audio src="carnarres.rm" system-language="es"/> -->
       <audio src="carnarren.rm"/>
     </switch>


<!-- AD -->

     <switch>
       <audio src="cardescen.rm" system-language="en"/>
       <audio src="cardescde.rm" system-language="de"/>
 <!--  <audio src="cardesces.rm" system-language="es"/> -->
       <audio src="cardescen.rm"/>
     </switch>

<!-- CAPTIONS -->

     <switch>
       <textstream src="carcapsen.rt" region="textregion" system-language="en" system-captions="on"/>
       <textstream src="carcapsde.rt" region="textregion" system-language="de" system-captions="on"/>
 <!--  <textstream src="carcapses.rt" region="textregion" system-language="es" system-captions="on"/> -->
       <textstream src="carcapsen.rt" region="textregion" system-captions="on"/>
     </switch>
    </par>
   </body>
</smil>
```

```
<smil>
 <head>
                <meta name="title" content="Elevator"/>
                <meta name="author" content="NCAM/NOVA"/>
                <meta name="copyright" content="©1998 WGBH Educational Foundation" />
   <layout>

<!-- define boxes here -->

   <root-layout background-color="black" height="180" width="200"/>
   <region id="videoregion" background-color="black" top="5" left="20" height="150" width="200"/>
   <region id="textregion" top="130" left="5" height="40" width="200"/>
  </layout>
 </head>

<!-- specify source files here -->

    <body>
     <par>
     <video src="elevator.rm" region="videoregion"/>
     <textstream src="elevcaps.rt" region="textregion"/>
     <audio src="elevdesc.rm"/>
     </par>
    </body>
</smil>
```

```
<window bgcolor="black" wordwrap="true">

<font size="3" face="times new roman" color="#FFFFFF">

<center>

<time begin="00:01.7"/>
Someone watching a car<BR>
accelerate toward light speed

<time begin="00:05.0"/><clear/>
would see something<br>
very strange.

<time begin="00:07.4"/><clear/>
It would seem as though the<br>
car itself were getting shorter

<time begin="00:10.5"/><clear/>
and that time for the person<br>
in the car was slowing down.

<time begin="00:14.4"/><clear/>
However, you wouldn't<br>
see these effects

<time begin="00:17.3"/><clear/>
until the car began to<br>
approach the speed of light.

<time begin="00:20.6"/><clear/>
At 90% of the speed of light

<time begin="00:23.0"/><clear/>
the car would appear to shrink<br>
to 44% of its usual length.

<time begin="00:28.6"/><clear/>
This thought experiment

<time begin="00:30.2"/><clear/>
answered Einstein's<br>
old question

<time begin="00:32.5"/><clear/>
about what he would see

<time begin="00:33.9"/><clear/>
```

```
if he traveled along<br>
with a beam of light.

<time begin="00:36.3"/><clear/>
He simply couldn't<br>
make the trip

<time begin="00:38.4"/><clear/>
for at the speed of light,<br>
length would contract to zero

<time begin="00:42.1" end="00:44.5"/><clear/>
and time would stop.

</center>

</font>

</window>
```

```
<window width="190" height="120" bgcolor="black" wordwrap="false" duration="01:23.3">

<font size="3" face="helvetica" color="#FFFFFF">

<center>

<time begin="00:01.1"/>
In one of Einstein's<br>
famous thought experiments

<br><time begin="00:04.2"/><clear>
he realized that<br>
gravity and acceleration

<br><time begin="00:07.1" end="00:09.6"/><clear>
appear to be<br>
the same phenomenon.

<br><time begin="00:011.2"/><clear>
Think about<br>
what would happen

<br><time begin="00:12.4"/><clear>
if an elevator suddenly<br>
went into a free-fall.

<br><time begin="00:15.6" end="00:19.2"/><clear>
The person inside would<br>
effectively weigh nothing.

<br><time begin="00:25.3"/><clear>
Next, imagine that<br>
same person

<br><time begin="00:27.05"/><clear>
in a motionless rocketship,<br>
so far from earth

<br><time begin="00:30.6"/><clear>
that the force of gravity<br>
is practically zero.

<br><time begin="00:33.3"/><clear>
Just like in the<br>
free-falling elevator

<br><time begin="00:36.5" end="00:38.6"/><clear>
he would also weigh nothing.
```

```
<br><time begin="00:41.3" end="00:44.4"/><clear>
Now, put the rocket in motion.

<br><time begin="00:44.5"/><clear>

<br><time begin="00:48.4"/><clear>
As the rocket speeds up

<br><time begin="00:50.4"/><clear>
the passenger's<br>
weight increases.

<br><time begin="00:52.4"/><clear>
To him, it feels<br>
just as if gravity

<br><time begin="00:55.8" end="00:58.5"/><clear>
had planted his feet<br>
firmly on the floor.

<time begin="01:07.9"/><clear>
Einstein realized<br>
that the force of gravity

<br><time begin="01:10.1"/><clear>
is actually the<br>
acceleration you feel

<br><time begin="01:12.6" end="01:15.2"/><clear>
as you move<br>
through space-time.

</center>

</font>

</window>
```

Elevator Transcript

Description:
An animation of an elevator.

Narrator:
In one of Einstein's famous thought experiments, he realized that gravity and acceleration appear to be the same phenomenon.

Description:
Einstein stands in the elevator.

Narrator:
Think about what would happen if an elevator suddenly went into a free-fall. The person inside would effectively weigh nothing.

Description:
Einstein floats in mid-air. Around him the elevator transforms into a rocketship's interior.

Narrator:
Next, imagine that same person in a motionless rocketship, so far from earth that the force of gravity is practically zero. Just like in the free-falling elevator, he would also weigh nothing.

Description:
In an exterior view, the rocketship floats in space.

Narrator:
Now, put the rocket in motion.

Description:
The rocket's engine fires. Einstein falls to the floor.

Narrator:
As the rocket speeds up, the passenger's weight increases. To him, it feels just as if gravity had planted his feet firmly on the floor.

Description:
From a squatting position, Einstein slowly rises. He steps backwards to regain his balance. The rocketship transforms back into an elevator.

Narrator:
Einstein realized that the force of gravity is actually the acceleration you feel as you move through space-time.

Description:
The elevator door swings open, allowing Einstein to exit.

Please use the BACK, GO BACK or PREVIOUS PAGE function of your browser to return to the previous slide.

WBGY | WGBH Educational Foundation | www.wgbh.org

Mark-up example for Checkpoint 1.1e: use of equivalent content in an APPLET

And the following code would produce whichever version you wanted or that your browser was capable of showing:

```
<APPLET code="Blink.class" width="500" height="40"
alt="Java applet: dancing text.">
If you were using a Java-enabled browser, you would see the text "It was the best of times, it was the worst of times..." dancing
across the screen, instead of this paragraph.
</APPLET>
```

◀ Example for Checkpoint 1.1e

Long description for Example 1.1

The graph shows Ice Cube Tray sales in July, August, September and October for the Far North, Near North, Near South, Deep South Regions. The data show two major trends. The first: that in any month, sales of ice cube trays increase the further south you go. The second: sales in each region decrease as the season changes from summer to autumn.

Or, if you want more detail:

- In the Far North, sales were 3, 4, 2, and 1 ice cube trays from July through October. Total sales in the Far North were 10 trays.
- In the Near North, sales were 8, 7, 5 and 2 ice cube trays from July through October. Total sales in the Near North were 22 trays.
- In the Near South, sales were 23, 18, 11 and 5 ice cube trays from July through October. Total sales in the Near South were 57 trays.
- In the Deep South, sales were 49, 43, 30 and 10 ice cube trays from July through October. Total sales in the Deep South were 132 trays.

◀ Back to example for Checkpoint 1.1a .